



本书用到的软件和开发包的下载教学视频
免费赠送
30小时
的Java教学视频

- ◆ 比Struts 2官方文档还要正确的知识体系讲解
- ◇ 本书最大的特点：**深入 详尽 解惑**
- ◆ 秉承作者一贯的让读者知其然而知其所以然的写作思路
- ◇ 循序渐进的**讲解**、恰如其分的**示例**、完美的知识**演绎**

Struts 2

深入详解

孙鑫 著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>



Sun Xin's Series

孙鑫作品系列

Java技术大系

Struts 2

深入详解

孙鑫 著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

2006年, WebWork 与 Struts 这两个优秀的 Java Web 框架的开发团队, 开始合作开发一个新的、结合了 WebWork 与 Struts 的优点, 并且更加优雅、扩展性更强的框架, 命名为“Struts 2”, 并于 2007 年上半年推出了 Struts 2 的正式版本。本书讲解的是 Struts 2 中号称“最佳可用”的 2.0.11 版本。

本书最大的特点就是深入、详尽、解惑。

深入 本书深入剖析了 Struts 2 内部的运行机制, 在知识的讲解上, 结合了 Web 程序的运行机制与 Struts 2 框架的内部工作机制, 以帮助读者更深入地掌握基于 Struts 2 的 Web 开发。

详尽 本书内容详尽, 几乎涵盖了 Struts 2 中所有的知识点。本书力求在每一个知识点的讲解上都清晰明了, 而不是简单地一带而过, 让读者对所学的知识理解得更为透彻。

解惑 本书秉承作者一贯的让读者知其然而知其所以然的写作风格, 在知识的讲解上, 不光让读者知道怎么用这个知识, 还让读者知道为什么要这么用。在问题的分析解决上, 从根源入手, 帮助读者了解问题发生的根本原因, 培养读者发现问题、分析问题、解决问题的能力。

本书语言生动、通俗易懂、讲解细致, 手把手地教你学习完成示例程序。全书给出了大量的示例, 相信通过这些示例的学习, 读者一定可以更好、更快地掌握 Struts 2 的应用开发。

本书介绍了 Struts 2 最新的注解(Annotation)配置方式, 这在国内外的同类图书中尚属首次。本书修正了 Struts 2 官方文档中的不少错误, 并针对 Struts 2 本身的一些 Bug 提供了解决方案。

本书可以作为 Struts 2 的学习用书, 也可以作为从事 Struts 2 开发的程序员的参考用书和必备手册, 还可以作为 Struts 2 官方文档的有益补充。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目(CIP)数据

Struts 2 深入详解 / 孙鑫著. —北京: 电子工业出版社, 2008.7
(Java 技术大系)
ISBN 978-7-121-06812-6

I. S… II. 孙… III. 软件工具—程序设计 IV. TP311.56

中国版本图书馆 CIP 数据核字(2008)第 077768 号

责任编辑: 葛 娜

印 刷: 北京市通州大中印刷厂

装 订: 三河市鹏成印业有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 43.25 字数: 921 千字

印 次: 2008 年 7 月第 1 次印刷

印 数: 5000 册 定价: 79.00 元(含光盘 1 张)

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前言

自从 2001 年 6 月, Struts 1.0 正式发布以来, Struts 就牢牢占据了 Web 框架市场老大的地位。然而, 随着软件开发技术的发展, Web 开发需求的变化, Struts 设计上的缺陷逐渐显露出来, 越来越多包含新的设计思想的 Web 开发框架涌现出来, 其中有代表性的包括: WebWork、Tapestry、Spring MVC。Struts 的开发团队意识到了 Struts 的问题, 于是和另一个优秀的 Web 框架——WebWork 的开发团队合作, 共同开发了一个新的、整合了 WebWork 与 Struts 的优点, 并且更加优雅、扩展性更强的框架, 命名为“Struts 2”, 原先 Struts 1.x 版本的产品称为“Struts 1”。

Struts 2 是在 WebWork 2 的基础上进行开发的, Struts 2.0 其实就是 WebWork 2.3, 它和 Struts 1 并没有关系, 如果读者使用过 WebWork 框架, 那么学习 Struts 2 就能很快上手; 相反, Struts 1 的用户学习 Struts 2, 面对的将是一个全新的框架。

Struts 2 的潜在用户包括了 Struts 1 和 WebWork 的用户, 相信随着时间的推移, Struts 2 还将续写 Struts 1 的辉煌。因此, 我们现在花点时间来学习 Struts 2, 可以提前为我们以后的开发工作打下基础, 提高我们的竞争力。

本书面向的读者

本书面向的读者群包括:

- 具有一定 Web 开发经验的开发人员;
- 具有其他 Web 框架使用经验, 想要学习 Struts 2 开发的开发人员;
- 正在从事 Java Web 开发的开发人员。

本书可以作为 Struts 2 的学习用书, 也可以作为从事 Struts 2 开发的程序员的参考用书和必备手册, 还可以作为 Struts 2 官方文档的有益补充。

本书的内容组织

本书在内容的安排上独具匠心, 在知识体系的讲解上由浅入深、循序渐进。从第 1 章的“认识 Struts 2”开始, 到第 5 章的“Struts 2 配置详解”, 逐步帮助读者熟悉 Struts 2, 了解 Struts 2 的开发流程, 掌握 Struts 2 的内部结构和 Struts 2 的详细配置。从第 6 章开始, 我们以一个在线用户注册程序为主线, 将 Struts 2 分散的知识有机地结合到用户注册程序的开发中, 让读者在一个真实的应用场景中一步一步掌握 Struts 2 的应用开发。本书还介绍了目前较热的 Ajax 开发技术, 以及 Struts 2 对 Ajax 的支持。最后本书还介绍了 Struts 2 与 Hibernate 和 Spring 的集成开发, 以及如何在 Struts 2 中开发 Struts 1 的 Web 应用。

本书给出了大量的示例, 讲解了一些开发技巧, 并针对一些问题的发生, 详细讲解了如何分析问题、查找问题和解决问题, 帮助读者培养发现问题、分析问题和解决问题的能力。

本书所有示例的完成都是采用分步进行的, 读者如果能够按照书中给出的开发步骤一步一步完成所有章节的示例程序, 相信一定能更好地掌握 Struts 2 的应用开发。

本书修正了 Struts 2 官方文档中的一些错误,并针对 Struts 2 本身的一些 Bug 提供了解决方案。

本书的示例程序

作者在编写本书时,使用的操作系统是 Windows XP Service Pack 2, JDK 版本是 1.6.0_05, Web 容器是 Tomcat 6.0.16, 开发工具是 MyEclipse 6.0.1 GA, 数据库是 MySQL 5.0.51a。

本书所有的示例程序在上述环境中都运行正常。

本书的示例程序是按章节组织的,每一章的示例程序都在 chXX 目录中(XX 表示章节序号),例如第 1 章的示例程序在 ch01 目录中。全书的示例程序在本书配套光盘中的 code 目录下可以找到。

本书示例程序的使用有两种方式:一种是将 chXX 目录下的 WebRoot 目录复制到 Tomcat 安装目录下的 webapps 子目录中,然后将 WebRoot 改名为 chXX,启动 Tomcat 服务器就可以访问这一章的 Web 程序了。例如,要访问第 10 章的 Web 应用程序,首先在本书配套光盘中找到 ch10 目录,然后将该目录下的 WebRoot 目录整体复制到 Tomcat 安装目录下的 webapps 子目录中,将 WebRoot 目录改名为 ch10,启动 Tomcat 服务器,参照第 10 章的内容,在浏览器的地址栏中输入 `http://localhost:8080/ch10/resource.action` 访问第 10 章的 Web 程序。

另一种方式是将每一章的项目导入 MyEclipse 的开发环境中,在 MyEclipse 中发布并访问每一章的 Web 程序。具体方式请参看第 8 章的 8.5 节。



注意

用户注册程序和第 17 章的 Web 程序的运行需要将 MySQL 的 JDBC 驱动复制到 Tomcat 安装目录的 lib 子目录中。

学习建议

本书前后内容的相关性比较强,所以建议读者循序渐进,一章一章地学习,同时在学习完每一章内容的时候,一定要试着动手自己实现本章的示例程序。当然,有经验的读者也可以选择某些章节的内容来学习。

本书的配套光盘说明

本书的配套光盘提供了两个部分的内容:本书所有实例的代码、价值人民币 330 元的 Java 教学视频(Flash 格式)。

(1) 实例代码

配套光盘中的实例代码是按照章节的顺序组织的,例如,第 7 章的例子程序位于光盘中的 code\ch07 目录下。用户注册程序和第 17 章的 Web 程序的运行需要将 MySQL 的 JDBC 驱动复制到 Tomcat 安装目录的 lib 子目录中。

(2) Java 教学视频

本套教学视频由作者本人录制，市场定价为人民币 330 元。为了帮助广大 Java 爱好者能够快速地步入 Java 企业级的开发，作者花费了大量的时间和精力将 AVI 格式的视频转换为 Flash 格式，免费提供给读者使用，希望读者能够从中受益。

这套视频的内容简介如下：

Lesson1: Java 的一些基本概念，Java 的起源，我们为什么要学习 Java 语言，Java 跨平台原理剖析；Java 环境变量的配置，初学者常犯的问题，Java 语言的基本数据类型和 Java 的语句。

Lesson2: Java 面向对象编程，面向对象编程和面向过程编程的区别，我们如何才能掌握面向对象的编程，类和对象的关系；讲解了 Java 中的构造、重载、this 和 super 变量、静态变量、Java 中的常量、继承、多态。

Lesson3 上: Java 中包的概念，类的说明符、方法的说明符、对象的销毁（JVM 垃圾回收器的演示），Java 中接口理解与掌握。

Lesson3 下: Java 中内部类的理解、运用与掌握，Java 中的异常处理。

Lesson4 上: Java 的常用包，“==”和“equals”的用法，基本数据类型与引用类型，对象的克隆。

Lesson4 下: 数组的复制与排序，Runtime 类与 Process 类，Class 类与反射 API，Runtime 类与单例设计模式。

Lesson5: Java 的多线程，程序、进程和线程的概念，实现多线程的两种方式，线程同步的原理，线程的死锁，运用 wait 和 notify 来实现 producer - consumer 关系，线程终止的两种情况。

Lesson6: Java 中集合类的讲解，栈、队列、链表等数据结构的讲解。

Lesson7: Java I/O 操作，字节流 InputStream 和 OutputStream，字符流 Reader 和 Writer，Java I/O 库的设计原则，字符集的编码，RandomAccessFile 类，管道流，对象的序列化。

Lesson8: Java 图形界面编程，AWT、布局管理器、事件模型，JFC、Swing 编程。应用 JBuilder 快速开发图形界面程序。

Lesson9: Applet 编程，详细讲解 Applet 的生命周期，子体的设置，Applet 从网页中获取信息，Applet 和浏览器的通信，更新显示的原理，用 Applet 来制作动画程序，以及动画程序的优化。

Lesson10: Java 网络编程，基于 TCP 的套接字编程，基于 UDP 的套接字编程，URL 和 URI，一个实用的下载程序。

关于封面漫画头像

封面的漫画头像由王裕民绘制，在此表示感谢。

最后，衷心地希望本书能够给读者带来知识，带来阅读上的快乐，读者对本书的肯定就是笔者最大的欣慰。由于本书的内容较多、涉及的技术较广，错误和疏漏之处在所难免，欢迎广大技术专家和读者指正。作者的联系方式是 csunxin@sina.com，读者也可以上作者的网站发表意见，网址是：<http://www.sunxin.org/>。

目 录

第 1 章 认识 Struts 2.....1	第 4 章 Struts 2 架构剖析..... 57
1.1 什么是 Web 框架.....1	4.1 XWork 2——Struts 2 的基础..... 57
1.1.1 模型 1.....2	4.2 Struts 2 的体系结构..... 58
1.1.2 模型 2 和 MVC.....5	4.3 Struts 2 框架的调用流程..... 61
1.1.3 Web 框架的诞生.....9	4.4 小结..... 62
1.2 从 Struts 1 到 Struts 2.....11	第 5 章 Struts 2 配置详解..... 63
1.2.1 其他 Web 框架.....11	5.1 Struts 2 的配置文件..... 63
1.2.2 Struts 1 的缺点.....12	5.1.1 web.xml..... 64
1.2.3 市场与技术的结合—— Struts 2 的诞生.....14	5.1.2 struts.properties..... 65
1.3 准备开发环境.....14	5.1.3 struts-default.xml..... 69
1.4 小结.....23	5.1.4 struts.xml..... 70
第 2 章 Struts 2 初体验.....24	5.1.5 struts-plugin.xml..... 74
2.1 下载和安装 Struts 2.....24	5.1.6 静态内容..... 75
2.2 使用 Subversion 来获取 Struts 2 的源代码.....25	5.2 struts.xml 配置详解..... 75
2.3 Hello World 程序.....28	5.2.1 Bean 配置..... 75
2.4 Struts 2 项目开发总结.....38	5.2.2 常量 (Constant) 的配置..... 76
2.5 小结.....38	5.2.3 包 (Package) 配置..... 78
第 3 章 由“登录”开始.....39	5.2.4 名称空间 (Namespace) 配置..... 80
3.1 使用领域对象接收用户输入.....39	5.2.5 包含 (Include) 配置..... 81
3.2 使用 ModelDriven action.....44	5.2.6 拦截器 (Interceptor) 配置..... 83
3.3 使用 action 的属性接收用户输入.....46	5.2.7 Action 配置..... 86
3.4 访问 request、session 和 application 对象.....47	5.2.8 Result 配置..... 86
3.4.1 与 Servlet API 解耦的 访问方式.....47	5.2.9 异常 (Exception) 配置..... 86
3.4.2 与 Servlet API 耦合的 访问方式.....52	5.3 配置 Action..... 86
3.5 小结.....56	5.3.1 Action 映射..... 86
	5.3.2 使用 method 属性..... 87
	5.3.3 动态方法调用..... 90
	5.3.4 模拟 Struts 1 的 ForwardAction..... 90
	5.3.5 默认的动作..... 91

5.3.6 通配符映射	92	7.3.2 操作符	142
5.4 配置 Result	95	7.3.3 访问 JavaBean 的属性	143
5.4.1 结果映射	95	7.3.4 调用方法	143
5.4.2 结果类型	97	7.3.5 调用静态方法和静态 字段	143
5.4.3 常用结果类型	99	7.3.6 调用构造方法	144
5.4.3.1 dispatcher 结果类型	99	7.3.7 索引访问	144
5.4.3.2 redirect 结果类型	100	7.3.7.1 数组和列表索引	144
5.4.3.3 redirectAction 结果 类型	101	7.3.7.2 JavaBean 的索引 属性	144
5.4.3.4 plainText 结果类型	103	7.3.7.3 OGNL 对象索引 属性	145
5.4.4 全局结果	103	7.3.8 对集合进行操作	145
5.4.5 动态结果	104	7.3.8.1 创建集合	145
5.4.6 在结果配置中使用参数	106	7.3.8.2 集合的伪属性	146
5.5 异常处理	107	7.3.8.3 投影 (projection)	147
5.6 使用注解配置 Action	110	7.3.8.4 选择 (selection)	147
5.6.1 与 Action 配置相关的 注解	111	7.3.9 lambda 表达式	148
5.6.2 使用注解配置 Action 的 注意事项	112	7.4 OGNL 的类型转换	148
5.6.3 使用注解配置 Action 的 实例	113	7.5 Struts 2 在 OGNL 基础上的 增强	149
5.7 小结	115	7.5.1 值栈 (ValueStack)	149
第 6 章 准备用户注册程序	116	7.5.2 [N]语法	150
6.1 创建数据库表	116	7.5.3 top 关键字	151
6.2 编写用户注册程序	118	7.5.4 访问静态成员	151
6.3 解决中文乱码问题	129	7.5.5 值栈中的 Action 实例	151
6.4 Struts 2 对依赖注入的支持	130	7.5.6 Struts 2 中的命名对象	151
6.4.1 Spring 插件	130	7.6 与 JSP 2.1 的冲突	152
6.4.2 在用户注册程序中集成 Spring	134	7.7 OGNL 表达式的综合应用	152
6.5 小结	136	7.8 小结	165
第 7 章 Struts 2 中的 OGNL	137	第 8 章 Struts 2 的标签库	166
7.1 为什么使用 OGNL	137	8.1 引用 Struts 2 的标签库	167
7.2 OGNL 基础	138	8.2 标签的语法	167
7.3 OGNL 表达式	141	8.3 通用标签	168
7.3.1 常量	141	8.3.1 数据标签	168
		8.3.1.1 property 标签	168
		8.3.1.2 set 标签	169

8.3.1.3	push 标签	170	8.4.2.10	radio 标签	218
8.3.1.4	param 标签	172	8.4.2.11	checkbox 标签	219
8.3.1.5	bean 标签	173	8.4.2.12	checkboxlist 标签	220
8.3.1.6	action 标签	174	8.4.2.13	doubleselect 标签	221
8.3.1.7	include 标签	177	8.4.2.14	combobox 标签	226
8.3.1.8	url 标签	178	8.4.2.15	optiontransferselct 标签	228
8.3.1.9	i18n 和 text 标签	181	8.4.2.16	updownselect 标签	231
8.3.1.10	date 标签	184	8.4.2.17	hidden 标签	233
8.3.1.11	debug 标签	186	8.4.2.18	submit 标签	233
8.3.2	控制标签	187	8.4.2.19	reset 标签	235
8.3.2.1	if/elseif/else 标签	188	8.4.2.20	label 标签	235
8.3.2.2	iterator 标签	189	8.4.2.21	token 标签	236
8.3.2.3	append 标签	191	8.4.2.22	file 标签	236
8.3.2.4	merge 标签	192	8.4.2.23	head 标签	236
8.3.2.5	generator 标签	194	8.4.3	非表单标签	237
8.3.2.6	subset 标签	197	8.4.3.1	component 标签	237
8.3.2.7	sort 标签	199	8.4.3.2	a 标签	240
8.4	UI 标签	200	8.4.3.3	actionerror、 actionmessage 和 fielderror 标签	240
8.4.1	模板和主题	201	8.5	使用 Struts 2 的表单标签改写 用户注册程序中的注册页面	242
8.4.1.1	Struts 2 支持的模板 引擎	201	8.6	小结	247
8.4.1.2	加载模板	201	第 9 章	国际化	248
8.4.1.3	选择主题	202	9.1	国际化与本地化	248
8.4.1.4	创建定制的主题	203	9.1.1	Locale	249
8.4.1.5	Struts 2 内置的 4 种 主题	203	9.1.2	资源包	251
8.4.2	表单标签	206	9.1.3	消息格式化	253
8.4.2.1	表单标签的公共 属性	206	9.2	Struts 2 对国际化的支持	254
8.4.2.2	表单标签的 name 和 value 属性	208	9.2.1	Struts 2 资源包的组织和 加载方式	255
8.4.2.3	ID 名分配	209	9.2.2	在消息文本中使用参数	255
8.4.2.4	form 标签	209	9.2.3	访问国际化消息	256
8.4.2.5	textfield 标签	210	9.3	将用户注册程序国际化	258
8.4.2.6	password 标签	211	9.4	错误消息的国际化	262
8.4.2.7	textarea 标签	211	9.5	Struts 2 设定访问用户的 locale 的流程	265
8.4.2.8	select 标签	212			
8.4.2.9	optgroup 标签	217			

9.6 为用户注册程序添加英文版本	266	第 12 章 对输入数据进行验证	325
9.7 小结	269	12.1 手动验证	326
第 10 章 拦截器	271	12.1.1 在 Action 的 execute() 方法中进行验证	326
10.1 拦截器的工作方式	271	12.1.2 在 validateXxx()方法中 进行验证	328
10.2 编写拦截器类	272	12.1.3 在 validate()方法中 进行验证	330
10.3 Struts 2 自带的拦截器	274	12.2 使用验证框架	332
10.4 配置拦截器	281	12.2.1 什么是验证器	332
10.5 PreResultListener 接口	282	12.2.2 注册验证器	332
10.6 开发执行安全验证的拦截器	285	12.2.3 开启验证	333
10.7 使用拦截器注解	290	12.2.4 定义验证规则	333
10.8 小结	294	12.2.5 验证文件的查找顺序	335
第 11 章 类型转换	295	12.2.6 Struts 2 内置的验证器	336
11.1 传统的数据处理过程	296	12.2.7 在用户注册程序中使用 验证框架	345
11.2 Struts 2 对类型转换的支持	296	12.2.8 确认密码的验证	349
11.2.1 使用 OGNL 表达式命名 参数	297	12.2.9 短路验证	350
11.2.2 Struts 2 内置的类型 转换器	297	12.2.10 开发自己的验证器	353
11.2.3 null 属性的处理	298	12.2.11 使用 visitor 字段验证器 复用验证	363
11.2.4 对 List、Map 和 Set 的 支持	298	12.2.12 同时使用验证框架和 手动验证	367
11.2.4.1 使用 List 来保存 图书信息的实例	298	12.2.13 客户端 JavaScript 验证	368
11.2.4.2 使用 Map 来保存 图书信息的实例	303	12.3 使用验证注解	369
11.2.5 使用 Set 来保存图书 信息的实例	307	12.3.1 与验证相关的注解	369
11.3 创建自定义的类型转换器	311	12.3.2 在用户注册程序中使用 验证注解	379
11.4 类型转换错误的处理	318	12.4 小结	383
11.5 使用类型转换注解	321	第 13 章 使用 FreeMarker 作为 表现层	384
11.5.1 TypeConversion 注解	321	13.1 FreeMarker 简介	385
11.5.2 Conversion 注解	322	13.1.1 FreeMarker 是什么	385
11.5.3 Element 注解	323	13.1.2 为什么要使用 FreeMarker	386
11.5.4 Key 注解	323	13.2 FreeMarker 的工作原理	386
11.5.5 KeyProperty 注解	324		
11.5.6 CreateIfNull 注解	324		
11.6 小结	324		

13.2.1	模板+数据模型=输出	386	13.3.4.11	function 和 return 指令	425
13.2.2	数据模型	388	13.3.4.12	noparse 指令	426
13.2.2.1	scalar 变量	391	13.3.4.13	escape 和 noescape 指令	427
13.2.2.2	hash 和 sequence 变量	392	13.3.5	内置函数	429
13.2.2.3	方法变量	392	13.3.5.1	操作字符串	429
13.2.2.4	宏变量	392	13.3.5.2	操作数字	437
13.2.3	模板	392	13.3.5.3	操作日期	439
13.3	FreeMarker 的语法和操作	393	13.3.5.4	操作布尔值	441
13.3.1	准备工作	393	13.3.5.5	操作序列	441
13.3.2	表达式	395	13.3.5.6	操作散列	446
13.3.2.1	直接指定值	395	13.3.6	在模板中定义变量	447
13.3.2.2	访问变量	397	13.4	Struts 2 对 FreeMarker 的支持	449
13.3.2.3	字符串操作	399	13.4.1	FreeMarker 标签	449
13.3.2.4	序列操作	400	13.4.2	在 FreeMarker 标签中使用动态属性	450
13.3.2.5	散列操作	400	13.4.3	JSP 标签支持	450
13.3.2.6	算术运算	400	13.4.4	变量解析与内置变量	451
13.3.2.7	比较操作	401	13.4.5	访问 Servlet/JSP 范围对象	451
13.3.2.8	逻辑操作	402	13.4.6	类型转换与本地化	453
13.3.2.9	内置函数	402	13.5	在用户注册程序中使用 FreeMarker 来实现表现层	453
13.3.2.10	方法调用	403	13.6	小结	459
13.3.2.11	缺少值处理	403	第 14 章	使用 SiteMesh 控制页面布局	461
13.3.2.11	操作符	403	14.1	SiteMesh 简介	462
13.3.2.12	操作符优先级	405	14.2	下载并安装 SiteMesh	462
13.3.3	插值 (interpolation)	406	14.3	感受 SiteMesh	464
13.3.4	指令	407	14.4	SiteMesh 的工作原理	467
13.3.4.1	if/elseif/else 指令	408	14.5	SiteMesh 的标签库	468
13.3.4.2	switch/case/default/break 指令	409	14.5.1	decorator 标签库	469
13.3.4.3	list/break 指令	411	14.5.2	page 标签库	470
13.3.4.4	include 指令	412	14.6	在装饰器中获取原始页面的内容	470
13.3.4.5	import 指令和名称空间	414			
13.3.4.6	assign 指令	418			
13.3.4.7	global 指令	419			
13.3.4.8	local 指令	420			
13.3.4.9	setting 指令	420			
13.3.4.10	macro、nested 和 return 指令	421			

14.7	装饰器的组合应用	471	16.2.1	Struts 2 对文件下载的支持	515
14.8	装饰器映射器	475	16.2.2	在 Struts 2 中实现文件下载	516
14.9	Struts 2 对 SiteMesh 的支持	479	16.2.3	改进下载程序	518
14.10	在用户注册程序中使用 SiteMesh 进行页面布局	480	16.3	小结	521
14.10.1	使用 JSP 作为表现层	480	第 17 章	在 Struts 2 中实现 CRUD	522
14.10.2	使用 FreeMarker 作为表现层	484	17.1	准备工作	522
14.11	小结	488	17.2	文章分类的 CRUD 操作	528
第 15 章	避免表单重复提交与等待页面	489	17.3	小结	533
15.1	避免表单重复提交	489	第 18 章	Struts 2 对 Ajax 的支持	535
15.1.1	避免表单重复提交的实现原理	490	18.1	Ajax 快速入门	535
15.1.2	Struts 2 的实现方式	490	18.1.1	Ajax 简介	536
15.1.3	使用 TokenInterceptor	491	18.1.2	DOM	537
15.1.4	使用 TokenSession StoreInterceptor	493	18.1.3	XMLHttpRequest	539
15.2	自动显示等待页面	493	18.2	Ajax 应用开发	540
15.2.1	ExecuteAndWaitInterceptor 的工作机制	494	18.2.1	使用 XML 作为数据的载体	541
15.2.2	使用 ExecuteAndWaitInterceptor	495	18.2.2	使用 JSON 串作为数据的载体	551
15.3	小结	497	18.2.2.1	理解 JSON	552
第 16 章	文件上传和下载	498	18.2.2.2	下载 JSON 的 Java 实现	554
16.1	文件上传	498	18.2.2.3	下载 JSON 的 JavaScript 脚本	555
16.1.1	基于表单的文件上传	498	18.2.2.4	使用 JSON 作为图书信息的数据格式	555
16.1.2	文件上传格式分析	499	18.2.3	使用 Dojo 简化 Ajax 应用的开发	561
16.1.3	Struts 2 对文件上传的支持	501	18.2.3.1	Dojo 概述	561
16.1.4	在 Struts 2 中实现文件上传	502	18.2.3.2	下载 Dojo 工具包	565
16.1.5	对文件上传进行更多的控制	507	18.2.3.3	在 ch18 Web 程序中使用 Dojo	565
16.1.6	多文件上传	509	18.3	在 Struts 2 中使用 JSON 插件	570
16.2	文件下载	515	18.3.1	JSON 插件	570

18.3.1.1	JSON 插件概述	571
18.3.1.2	定制序列化和反序列化	571
18.3.1.3	下载 JSON 插件	572
18.3.2	使用 JSON 插件的实例	573
18.4	Struts 2 的 Ajax 标签	578
18.4.1	Ajax 标签的基础知识	579
18.4.1.1	共同的属性	579
18.4.1.2	等待指示符	579
18.4.1.3	事件发布/订阅模型	580
18.4.1.4	URL	580
18.4.1.5	阻止请求	580
18.4.2	使用 Ajax 标签前的准备工作	581
18.4.3	div 标签	583
18.4.4	submit 标签	593
18.4.5	a 标签	597
18.4.6	tabbedPanel 标签	599
18.4.7	autocompleter 标签	604
18.4.7.1	autocompleter 标签简介	604
18.4.7.2	准备工作	605
18.4.7.3	autocompleter 标签用法详析	608
18.4.8	tree 和 treenode 标签	614
18.4.8.1	用 tree 和 treenode 标签构建静态树	614
18.4.8.2	用 tree 标签构建动态树	617
18.5	Struts 2 对 DWR 的支持	623
18.6	小结	630

第 19 章 集成 Hibernate 和 Spring 631

19.1	Hibernate 简介	632
19.2	Hibernate 的下载和安装	632
19.3	Hibernate 快速入门	633
19.3.1	配置文件	633
19.3.2	主要接口和类	635
19.3.3	总结	638
19.4	在 Struts 2 中集成 Hibernate	638
19.5	集成 Spring	647
19.6	小结	651

第 20 章 在 Struts 2 中开发 Struts 1 的 Web 应用 652

20.1	Struts 1 插件	652
20.1.1	Struts 1 插件的工作原理	652
20.1.2	使用 Struts 1 插件	653
20.2	用户注册实例	656
20.2.1	准备工作	656
20.2.2	阶段 1: 简单的 Struts 1 Action	658
20.2.3	阶段 2: 带有 ActionForm 的 Struts 1 Action	660
20.2.4	阶段 3: 添加手动验证	664
20.2.5	阶段 4: 使用 Commons Validator 验证框架	665
20.3	小结	670

附录 A Struts 1 与 Struts 2 的对比 671

附录 B WebWork 2 与 Struts 2 的对比 673

附录 C 从 WebWork 2 移植到 Struts 2 675

认识 Struts 2

本章要点

- 掌握 MVC 架构模式
 - 理解 Web 框架是什么
 - 弄清楚 Struts 1、Struts 2 和 WebWork 三者之间的关系
 - 搭建本书的开发环境
-

Struts 2 已经推出了稳定的版本，其先进的架构和设计思想，随着时间的推移，相信会有越来越多的人开始使用 Struts 2。基于 Struts 原有的人气，加上采用了 WebWork 的设计架构，Struts 2 必将成为最为流行的 Web 开发框架之一。

1.1 什么是 Web 框架

随着 Internet 的发展，基于 HTTP 协议和 HTML 标准的 Web 应用呈几何数量级增长，人们在不知不觉中已经被网络悄悄地改变了生活。在网络普及之前，我们购买图书要去书店，给亲人汇钱要去邮局或者银行，而现在，一切都是这么便捷，你可以在网上购买图书、汇款、缴纳电话费，你甚者为远在他乡的女朋友订购一束玫瑰。各种各样的网上业务丰富了我们的生活，节省了我们的时间，提高了我们的工作效率，改善了我们的生活品质。

支撑这些网上业务的就是各种各样的 Web 应用，而这些 Web 应用又是用各种 Web 技术开发的。早期的 Web 应用主要是静态页面的浏览（如新闻的浏览），这些静态页面使用 HTML 语言来编写。随着网络的发展，很多线下业务开始向网上发展，基于 Internet 的 Web 应用也变得越来越复杂，不但要和数据库交互，而且还要和用户进行交互，由此衍生出了各种服务器端页面编写技术，目前应用得较多的三种服务器端页面编写技术就是 ASP、JSP 和 PHP。

JSP 通过在 HTML 页面文件中嵌入 Java 脚本代码，从而实现动态网页功能。HTML

标签负责页面的布局和显示外观, Java 代码负责提取动态数据, 实现业务逻辑。对于简单的应用, 采用这种方式可以简化开发难度, 加快开发进度; 然而, 对于一些较大的应用, 大量的 HTML 和 Java 代码的混杂导致了页面的显示逻辑和业务逻辑交错在一起, 使得代码的可读性变差, 维护难度增加, 不利于 HTML 编辑人员和 Java 开发人员的分工协作。

为了分离页面的表现逻辑和业务逻辑, 使程序各部分的职责更加清晰, 增强代码的可维护性, SUN 公司在 JSP 的早期规范中制定了两种 JSP 开发模型, 分别为模型 1 (Model 1) 和模型 2 (Model 2)。

1.1.1 模型 1

模型 1 使用 JSP+JavaBeans 技术将页面显示和业务逻辑处理分开。JSP 实现页面的显示, JavaBean 对象用来承载数据和实现业务逻辑。模型 1 的结构如图 1-1 所示。

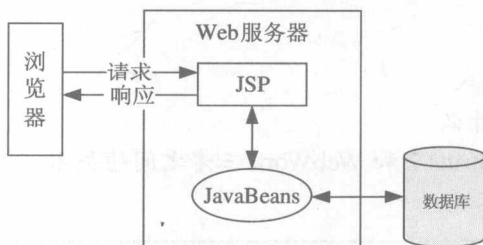


图 1-1 模型 1 的 JSP 架构

在模型 1 中, JSP 页面独自响应请求并将处理结果返回给客户, 所有的数据通过 JavaBean 来处理, JSP 实现页面的显示。

我们看一个按照模型 1 的结构实现的用户登录的例子。在这个例子中有两个 JavaBean 类, UserBean 和 UserCheckBean, UserBean 对象负责保存用户数据, UserCheckBean 对象负责实现对用户登录信息进行验证的业务逻辑。

UserBean 的代码片段如例 1-1 所示。

例 1-1 UserBean.java

```
package org.sunxin.struts2.ch01.model1;

import java.io.Serializable;

public class UserBean implements Serializable
{
    //用户名
    private String name;
    //密码
    private String password;

    ...
    //省略了 name 和 password 的 getter 和 setter 方法
}
```

UserCheckBean 的代码如例 1-2 所示。

例 1-2 UserCheckBean.java

```
package org.sunxin.struts2.ch01.model1;

public class UserCheckBean
{
    private UserBean user;

    public UserCheckBean()
    {
    }

    public UserCheckBean(UserBean user)
    {
        this.user=user;
    }

    ...
    //省略了 user 的 getter 和 setter 方法

    public boolean validate()
    {
        String name=user.getName();
        String password=user.getPassword();

        //此处为了简化而直接对用户名和密码进行了判断
        if("张三".equals(name) && "1234".equals(password))
        {
            return true;
        }
        else
        {
            return false;
        }
    }
}
```

此外，还有三个 JSP 页面：login.jsp、loginchk.jsp 和 welcome.jsp。login.jsp 负责显示登录表单，代码如例 1-3 所示。

例 1-3 login.jsp

```
<%@ page contentType="text/html;charset=GBK" %>
<html>
  <head><title>登录页面</title></head>
  <body>
    <form method="post" action="loginchk.jsp">
      用户名: <input type="text" name="name"><br>
```



```

        密码: <input type="password" name="password"><p>
        <input type="reset" value="重填">
        <input type="submit" value="登录">
    </form>
</body>
</html>

```

welcome.jsp 负责向用户显示欢迎信息，代码如例 1-4 所示。

例 1-4 welcome.jsp

```

<%@ page contentType="text/html;charset=GB2312" %>
<jsp:useBean id="user" scope="session"
    type="org.sunxin.struts2.ch01.model1.UserBean"/>

欢迎你, <jsp:getProperty name="user" property="name"/>!

```

loginchk.jsp 负责调用 UserCheckBean 对用户名和密码进行验证，如果验证通过，则转向欢迎页面；否则输出错误信息，代码如例 1-5 所示。

例 1-5 loginchk.jsp

```

<%@ page contentType="text/html;charset=GBK" %>
<%@ page import="org.sunxin.struts2.ch01.model1.UserCheckBean" %>

<%request.setCharacterEncoding("GB2312");%>

<jsp:useBean id="user" scope="session" class="org.sunxin.struts2.ch01.
model1.UserBean"/>
<jsp:setProperty name="user" property="*" />

<%
    UserCheckBean uc=new UserCheckBean(user);
    if(uc.validate())
    {
%>
<jsp:forward page="welcome.jsp"/>
<%
    }
    else
    {
        out.println("用户名或密码错误,请<a href=\"login.html\">重新登录</a>");
    }
%>

```

完整的代码请参看本书配套光盘中的 ch01 目录，测试时，输入 URL：

```
http://localhost:8080/ch01/model1/login.jsp
```

通过这个例子可以看到，模型 1 将页面显示和业务逻辑进行了分离，不足之处是在 JSP 页面中仍然需要编写流程控制和调用 JavaBean 的代码，当需要处理的业务逻辑非常复杂