



超过450分钟的Struts 2课堂教学视频  
提供配书视频对应的PPT、课堂讲义和源代码

Struts2的架构和运行流程  
Action、Result和Interceptor  
值栈的功能和使用OGNL  
Struts 2标签体系  
Struts 2对Ajax的支持  
Struts 2对文件上传下载的支持  
防止重复提交和进度条  
整合Spring、JFreeChart、SiteMeth  
Struts 2的国际化  
使用Struts 2的验证框架和类型转换  
Struts 2对零配置的支持

# Struts 2

陈臣 王斌 孙琳 王鸽 蒲蓬勃 编著



清华大学出版社



# Struts 2

陈 臣 王 斌 孙 琳 王 鸽 蒲蓬勃 编著

清华大学出版社  
北 京

## 内 容 简 介

Struts 2 是一套非常优秀的 Web 应用框架, 实现优雅、功能强大、使用简洁, 已经出现几年了。随着它被越来越多的开发人员掌握和使用, 已有大量的企业应用项目采用 Struts 2 来作为 Web 框架进行开发, 可以这么说: 掌握 Struts 2 的开发, 是如今每个进行 Web 开发的程序员的一个必备技能。本书通过浓缩作者多年的开发经验和教学经验, 对 Struts 2 的庞杂知识进行抽丝剥茧, 提炼出 Struts 2 的核心知识和重要技能, 让读者在短时间内理解和掌握 Struts 2, 迅速成为使用 Struts 2 开发的高手。

本书包含了 Struts 2 各个核心部分的功能, 以及 Struts 2 的国际化、验证框架、单元测试和类型转换等内容; 同时覆盖 Struts 2 和 Spring、FreeMarker、AJAX、DWR、JfreeChart、SiteMesh 等其他技术结合使用的知识; 还讲述了使用 Struts 2 来进行文件上传下载、进度条、防止重复提交和零配置等内容, 并给出一个实用的综合实例来示范如何使用 Struts 2 来进行应用开发。

本书的读者定位为具有一定 Web 开发基础知识(包括 HTML、JSP、Servlet、XML、JDBC 等基础知识)的开发人员, 也可以作为具备相应基础知识的高校学生深入学习 Struts 2 的参考读物。

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

研磨 Struts 2/ 陈臣等编著. —北京: 清华大学出版社, 2011.10

ISBN 978-7-302-25918-3

I. ①研… II. ①陈… III. ①软件工具——程序设计 IV. ①TP311.56

中国版本图书馆 CIP 数据核字 (2011) 第 115739 号

责任编辑: 栾大成

责任校对: 徐俊伟

责任印制: 王秀菊

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](mailto:jsjic@tup.tsinghua.edu.cn)

质 量 反 馈: 010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 刷 者: 清华大学印刷厂

装 订 者: 三河市金元印装有限公司

经 销: 全国新华书店

开 本: 188×260 印 张: 31.25 字 数: 687 千字

附 DVD1 张

版 次: 2011 年 10 月第 1 版 印 次: 2011 年 10 月第 1 次印刷

印 数: 1~4000

定 价: 59.00 元

# 前言

## 创作背景

Struts 2 是一套非常优秀的 Web 应用框架，实现优雅、功能强大、使用简洁。近几年来，随着它被越来越多的开发人员掌握和使用，已有大量的企业应用项目采用 Struts 2 来作为 Web 框架进行开发，可以这么说：掌握 Struts 2 的开发，是如今每个进行 Web 开发的程序员的一个必备技能。

在软件开发行业中，新知识层出不穷，程序员的日常工作也非常繁重，学习时间也有限，如何才能**快速地掌握** Struts 2 的知识，以适应工作的需要，并积累自己的技能，为今后的升职打下坚实的基础呢？

大学毕业的学生越来越多，他们找工作越来越困难，急需有内容实用、难度适中、重点难点突出的学习书籍，用以掌握 Struts 2 必备的技能，从而适应企业的实际需要。

到底要如何才能快速的学习和掌握 Struts 2 呢？

这正是创作本书的目的，通过浓缩编者多年的开发经验和教学经验，对 Struts 2 的庞杂知识进行抽丝剥茧，提炼出 Struts 2 的核心知识和重要技能，让读者在短时间理解 and 掌握 Struts 2，迅速成为使用 Struts 2 开发的高手。

## 本书内容

本书完整地包含了 Struts 2 各个核心部分的功能，以及 Struts 2 的国际化、验证框架、单元测试和类型转换等内容；同时覆盖 Struts 2、Spring、FreeMarker、AJAX、DWR、JfreeChart 和 SiteMesh 等其他技术结合使用的知识；还介绍了使用 Struts 2 来进行文件上传下载、进度条、防止重复提交和零配置等内容，并给出一个实用的综合实例来示范如何使用 Struts 2 来进行应用开发。

## 本书特色

- 采用速成培训的思路，本书就好比是快速学习的教材，如同由经验丰富的老师在指导学习。
- 把 Struts 2 的知识分成四个层次，分清主次构建知识体系。

第一部分是核心内容，必须掌握的，编者根据经验，把实际工作中最常用的 Struts 2 最核心的内容加以总结。包括第 1~8 章。

第二部分是常用内容，是工作中相对较为常用的知识，能掌握或者理解更好。包括第 9~11 章。

第三部分是选学内容，根据工作需要和时间来灵活安排，但是根据我们的经

验，使用到的概率还是很大的。包括第 13~15 章和 21 章。

第四部分是参考内容，可以当作工作中的工具书。包括第 16~20 章。

- 通过把 Struts 2 的知识进行分层梳理，指导读者以相对较小的付出掌握最重要的知识。

简而言之，本书特色就是：“快速”、“深入”、“实用”。

## 读者定位

本书的读者定位是具有一定 Web 开发基础知识(包括 HTML、JSP、Servlet、XML、JDBC 等基础知识)的开发人员，也可以作为具备相应基础知识的高校学生深入学习 Struts 2 的参考读物。

## 阅读指南

建议按照先后顺序和知识层次进行系统的学习。

- 重视理论知识

有些开发人员忽略理论知识的学习，愿意一上来就看代码，这是很不好的习惯。学习理论知识是很重要的，全面、深入、准确、系统地掌握理论知识，有利于梳理知识结构，整理出一张知识网，从而更好地把新知识融入到已有的知识体系中。

- 重视动手练习

建议读者在学习过程中跟着本书的进度编码测试和练习。只有亲自动手编码才能体会出其中的奥妙，否则容易流于表面，影响学习效果。

## 真诚致谢

首先，感谢清华大学出版社的员工，他们给予本书很多中肯的意见和建议，对本书从选题到出版的各个环节，都给予了大量的指导和帮助。

然后，感谢张开涛先生，他对本书的内容提出了很多有用的意见和建议。

再次，感谢我的家人在写作期间给予我的无限支持和关爱。

最后，提前感谢购买本书的朋友们，你们的信任和赏识是我们继续前进的动力，对于本书有任何意见或建议，可以直接与我们联系，联系邮件：Struts 2\_2011@yahoo.cn，我们也很乐意与各位朋友交流 Struts 2 或是其他相关的技术内容。

# 目 录

## 第 1 章 Struts 2 概述 .....1

### 1.1 MVC 模式.....2

#### 1.1.1 MVC 模式简介.....2

#### 1.1.2 MVC 模式基础.....2

#### 1.1.3 MVC 模式作用.....4

#### 1.1.4 Model2 模型中的 MVC .....5

### 1.2 Struts 2 基础 .....5

#### 1.2.1 Struts 1 时代 .....5

#### 1.2.2 Struts 2 出世 .....6

#### 1.2.3 Struts 2 基本组成 .....7

#### 1.2.4 Struts 2 能干什么 .....8

### 1.3 Struts 2 和 MVC .....9

## 第 2 章 Struts 2 的 HelloWorld.....11

### 2.1 概述.....12

### 2.2 准备开发环境和运行环境.....12

#### 2.2.1 配置 JDK .....12

#### 2.2.2 下载并安装 Eclipse.....13

#### 2.2.3 下载 Tomcat 并在 Eclipse 里面引用 Tomcat .....15

#### 2.2.4 下载 Struts 2 以及认识 Struts 2 包结构 .....17

### 2.3 从 Servlet+JSP+JavaBean 实现

#### MVC 开始.....17

#### 2.3.1 起点概述.....17

#### 2.3.2 建立动态的 Web 工程 .....18

#### 2.3.3 配置工程运行环境.....19

#### 2.3.4 编写 JavaBean .....20

#### 2.3.5 编写 Servlet .....22

#### 2.3.6 配置 web.xml.....23

#### 2.3.7 编写登录页面.....23

#### 2.3.8 编写欢迎页面.....25

#### 2.3.9 测试示例.....25

#### 2.3.10 中文处理.....26

### 2.4 Struts 2 的 HelloWorld .....28

#### 2.4.1 概述.....28

#### 2.4.2 配置 Struts 2 的运行环境..29

#### 2.4.3 实现动作类 (Action) .....30

#### 2.4.4 配置 struts.xml .....31

#### 2.4.5 制作登录页面 .....33

#### 2.4.6 制作欢迎页面 .....34

#### 2.4.7 测试示例 .....34

### 2.5 开发中的一些小窍门 .....35

#### 2.5.1 概述 .....35

#### 2.5.2 使用 DTD 来获得 XML 帮助 .....35

#### 2.5.3 在新建菜单中添加 JSP .....38

#### 2.5.4 获得 Action 的全类名 .....39

#### 2.5.5 获得 JSP 在 Web 工程中的 绝对路径 .....40

## 第 3 章 Struts 2 的架构和运行流程.43

### 3.1 Struts 2 的系统架构 .....44

#### 3.1.1 Struts 2 的系统架构 .....44

#### 3.1.2 各模块说明 .....44

### 3.2 Struts 2 的运行流程 .....45

### 3.3 Struts 2 的核心概念 .....50

### 3.4 小结 .....53

## 第 4 章 Action .....55

### 4.1 Action 基础 .....56

#### 4.1.1 Action 是什么 .....56

#### 4.1.2 Action 能干什么 .....56

#### 4.1.3 Action 的基本配置 .....57

### 4.2 Action 的实现 .....58

#### 4.2.1 POJO 的实现.....58

#### 4.2.2 实现 Action 接口 .....58

#### 4.2.3 继承 ActionSupport 类.....59

#### 4.2.4 execute 方法内部实现方式64

#### 4.2.5 简单的单元测试 .....66

### 4.3 Action 的数据 .....67

#### 4.3.1 数据来源 .....67

|                                            |     |                                       |     |
|--------------------------------------------|-----|---------------------------------------|-----|
| 4.3.2 基本的数据对应方式 .....                      | 67  | 5.5.3 在页面输出异常信息 .....                 | 116 |
| 4.3.3 传入非 String 类型的值 ....                 | 75  | 5.6 PreResultListener .....           | 118 |
| 4.3.4 如何处理传入多个值 .....                      | 79  | 5.6.1 什么是 PreResultListener .....     | 118 |
| 4.4 Action 的配置 .....                       | 81  | 5.6.2 PreResultListener<br>实现示例 ..... | 118 |
| 4.4.1 <package> 的配置 .....                  | 81  | 5.7 自定义 Result .....                  | 120 |
| 4.4.2 <action> 的配置 .....                   | 83  | 5.7.1 自定义 Result 概述 .....             | 120 |
| 4.4.3 分模块配置方式 .....                        | 83  | 5.7.2 开发自定义 Result .....              | 121 |
| 4.4.4 使用通配符 .....                          | 86  |                                       |     |
| 4.4.5 默认类配置方式 .....                        | 87  |                                       |     |
| 4.5 Action 的其他重要知识 .....                   | 89  | <b>第 6 章 拦截器</b> .....                | 123 |
| 4.5.1 Action 的生命周期 .....                   | 89  | 6.1 拦截器基础 .....                       | 124 |
| 4.5.2 调用非 execute 方法 .....                 | 90  | 6.1.1 拦截器是什么 .....                    | 124 |
| <b>第 5 章 Result</b> .....                  | 93  | 6.1.2 拦截器的优点 .....                    | 124 |
| 5.1 Result 基础 .....                        | 94  | 6.1.3 拦截器有什么 .....                    | 125 |
| 5.1.1 Result 是什么和能干什么 .....                | 94  | 6.2 预定义的拦截器 .....                     | 125 |
| 5.1.2 Result 和 ResultType .....            | 94  | 6.2.1 预定义的拦截器 .....                   | 125 |
| 5.1.3 Result 有什么 .....                     | 94  | 6.2.2 预定义的拦截器栈 .....                  | 130 |
| 5.2 预定义的 Result .....                      | 95  | 6.2.3 预定义拦截器的配置<br>使用 .....           | 132 |
| 5.2.1 预定义的 ResultType .....                | 95  | 6.2.4 拦截器的调用顺序 .....                  | 133 |
| 5.2.2 如何配置使用 .....                         | 96  | 6.2.5 实践一下 .....                      | 135 |
| 5.2.3 名称为 dispatcher 的<br>ResultType ..... | 98  | 6.3 自定义的拦截器 .....                     | 140 |
| 5.2.4 名称为 redirect 的<br>ResultType .....   | 100 | 6.3.1 什么是自定义的拦截器 .....                | 140 |
| 5.2.5 名称为 chain 的<br>ResultType .....      | 102 | 6.3.2 开发自定义拦截器 .....                  | 141 |
| 5.2.6 名称为 FreeMarker 的<br>ResultType ..... | 105 | 6.3.3 向拦截器传入参数 .....                  | 145 |
| 5.2.7 其他 ResultType .....                  | 107 | 6.4 两个有用的拦截器 .....                    | 148 |
| 5.3 全局 Result .....                        | 107 | 6.4.1 更强大的 logger 拦截器 .....           | 148 |
| 5.3.1 全局 Result 概述 .....                   | 107 | 6.4.2 登录检查拦截器 .....                   | 151 |
| 5.3.2 配置和使用 .....                          | 107 | <b>第 7 章 值栈和 OGNL</b> .....           | 155 |
| 5.3.3 搜寻 Result 的顺序 .....                  | 108 | 7.1 值栈 .....                          | 156 |
| 5.3.4 应用示例 .....                           | 108 | 7.1.1 值栈是什么 .....                     | 156 |
| 5.4 使用通配符 .....                            | 110 | 7.1.2 值栈能干什么 .....                    | 156 |
| 5.5 Struts 2 的异常映射 .....                   | 113 | 7.1.3 值栈有什么 .....                     | 156 |
| 5.5.1 异常映射基础 .....                         | 113 | 7.1.4 ActionContext 的<br>基本使用 .....   | 157 |
| 5.5.2 局部异常映射与全局<br>异常映射 .....              | 115 | 7.1.5 ValueStack 的基本使用 .....          | 158 |
|                                            |     | 7.2 OGNL .....                        | 161 |
|                                            |     | 7.2.1 OGNL 概述 .....                   | 161 |
|                                            |     | 7.2.2 OGNL 的基本使用 .....                | 161 |

|                                                  |     |                                 |     |
|--------------------------------------------------|-----|---------------------------------|-----|
| 7.2.3 访问静态方法和静态属性.....                           | 164 | 8.4.4 form 标签.....              | 211 |
| 7.3 使用 OGNL 访问复杂对象.....                          | 165 | 8.4.5 文本域型标签 .....              | 212 |
| 7.3.1 访问域对象.....                                 | 165 | 8.4.6 选择型标签 .....               | 214 |
| 7.3.2 访问 List 或数组 .....                          | 169 | 8.4.7 其他简单标签 .....              | 218 |
| 7.3.3 访问 Map .....                               | 171 | 8.4.8 复杂操作标签 .....              | 219 |
| 7.3.4 组合使用.....                                  | 172 | 第 9 章 国际化 .....                 | 227 |
| 7.4 ActionContext 和 ServletActionContext .....   | 173 | 9.1 概述 .....                    | 228 |
| 7.4.1 再探 ActionContext .....                     | 173 | 9.2 Java 内置的国际化 .....           | 228 |
| 7.4.2 ServletActionContext.....                  | 175 | 9.2.1 Java 内置的国际化入门 ..          | 228 |
| 7.4.3 ActionContext 和 ServletActionContext ..... | 177 | 9.2.2 对中文的支持 .....              | 230 |
| 第 8 章 Struts 2 的 Taglib.....                     | 179 | 9.2.3 向国际化信息传入参数 ..             | 231 |
| 8.1 Struts 2 的 Taglib 概述.....                    | 180 | 9.2.4 小结 .....                  | 233 |
| 8.1.1 概述.....                                    | 180 | 9.3 Struts 2 中的国际化入门 .....      | 233 |
| 8.1.2 Struts 2 标签的分类 .....                       | 181 | 9.3.1 引用国际化信息资源文件 .....         | 233 |
| 8.2 数据标签.....                                    | 181 | 9.3.2 在 JSP 中访问国际化信息 .....      | 234 |
| 8.2.1 property 标签 .....                          | 181 | 9.3.3 指定浏览器使用的语种 ..             | 234 |
| 8.2.2 set 标签 .....                               | 184 | 9.4 资源文件读取顺序 .....              | 237 |
| 8.2.3 push 标签 .....                              | 187 | 9.4.1 全局级资源文件 .....             | 237 |
| 8.2.4 bean 与 param 标签 .....                      | 188 | 9.4.2 包级资源文件 .....              | 238 |
| 8.2.5 date 标签 .....                              | 189 | 9.4.3 类级资源文件 .....              | 238 |
| 8.2.6 debug 标签.....                              | 190 | 9.4.4 资源文件的覆盖顺序 .....           | 239 |
| 8.2.7 url 与 a 标签 .....                           | 191 | 9.5 Struts 2 访问国际化信息的不同方式 ..... | 240 |
| 8.2.8 include 与 param 标签 .....                   | 193 | 9.5.1 用<s:text>访问国际化信息 .....    | 240 |
| 8.2.9 其他标签.....                                  | 194 | 9.5.2 用表单标签访问国际化信息 .....        | 242 |
| 8.3 控制标签.....                                    | 194 | 9.5.3 在 Action 中访问国际化信息 .....   | 243 |
| 8.3.1 if、elseif 与 else 标签 .....                  | 194 | 9.6 指定语言信息的不同方式 .....           | 244 |
| 8.3.2 iterator 标签 .....                          | 195 | 9.6.1 概述 .....                  | 244 |
| 8.3.3 append 与 param 标签 .....                    | 200 | 9.6.2 Struts 2 的常量设置 .....      | 244 |
| 8.3.4 generator 标签.....                          | 201 | 9.6.3 由用户选择语言 .....             | 244 |
| 8.3.5 merge 标签.....                              | 202 | 9.6.4 三种方式的覆盖关系 .....           | 246 |
| 8.3.6 sort 标签.....                               | 203 | 第 10 章 验证框架 .....               | 247 |
| 8.3.7 subset 标签.....                             | 206 | 10.1 验证框架的作用 .....              | 248 |
| 8.4 UI 标签 .....                                  | 208 |                                 |     |
| 8.4.1 概述.....                                    | 208 |                                 |     |
| 8.4.2 模板和主题.....                                 | 208 |                                 |     |
| 8.4.3 表单标签的通用属性.....                             | 210 |                                 |     |

|                                                                        |     |                                     |            |
|------------------------------------------------------------------------|-----|-------------------------------------|------------|
| 10.2 验证框架入门 .....                                                      | 248 | 10.7 验证器框架的查找顺序 .....               | 272        |
| 10.2.1 准备基础资源 .....                                                    | 248 | 10.7.1 验证器框架的查找顺序 .....             | 272        |
| 10.2.2 初会验证框架 .....                                                    | 251 | 10.7.2 验证器短路 .....                  | 274        |
| 10.2.3 验证框架的运行原理 .....                                                 | 253 |                                     |            |
| 10.3 验证器类型 .....                                                       | 257 | <b>第 11 章 类型转换 .....</b>            | <b>277</b> |
| 10.3.1 字段验证器 .....                                                     | 257 | 11.1 类型转换的作用 .....                  | 278        |
| 10.3.2 动作验证器 .....                                                     | 258 | 11.2 内建类型转换器 .....                  | 278        |
| 10.4 内建验证器 .....                                                       | 260 | 11.2.1 简单类型 .....                   | 278        |
| 10.4.1 概述 .....                                                        | 260 | 11.2.2 枚举类型 .....                   | 279        |
| 10.4.2 类型转换错误验证器<br>(conversion) .....                                 | 262 | 11.2.3 复合类型 .....                   | 280        |
| 10.4.3 整数验证器 (int) .....                                               | 262 | 11.3 自定义类型转换器 .....                 | 284        |
| 10.4.4 短整数验证器 (short) 和长<br>整数验证器 (long) .....                         | 262 | 11.3.1 概述 .....                     | 284        |
| 10.4.5 日期验证器 (date) .....                                              | 262 | 11.3.2 实现自定义类型转换器 .....             | 284        |
| 10.4.6 双精度浮点验证器<br>(double) .....                                      | 263 | 11.3.3 注册和引用自定义类型<br>转换器 .....      | 285        |
| 10.4.7 邮件验证器 (E-mail) .....                                            | 263 | 11.4 类型转换器引用方式 .....                | 287        |
| 10.4.8 URL 验证器 (url) .....                                             | 263 | 11.4.1 全局级类型转换器引用 .....             | 287        |
| 10.4.9 必填验证器 (required) .....                                          | 263 | 11.4.2 类级类型转换器引用 .....              | 287        |
| 10.4.10 必填字符串验证器<br>(requiredstring) .....                             | 263 | <b>第 12 章 CRUD 综合运用 .....</b>       | <b>289</b> |
| 10.4.11 字符串长度验证器<br>(stringlength) .....                               | 264 | 12.1 概述 .....                       | 290        |
| 10.4.12 正则表达式验证器<br>(regex) .....                                      | 264 | 12.1.1 功能简介 .....                   | 290        |
| 10.4.13 字段表达式验证器<br>(fieldexpression)<br>和表达式验证器<br>(expression) ..... | 264 | 12.1.2 使用的技术 .....                  | 290        |
| 10.4.14 访问者验证器<br>(visitor) .....                                      | 265 | 12.1.3 准备开发环境 .....                 | 291        |
| 10.5 自定义验证器 .....                                                      | 266 | 12.2 实现域对象和数据层 .....                | 291        |
| 10.5.1 概述 .....                                                        | 266 | 12.2.1 建表和域对象 .....                 | 291        |
| 10.5.2 实现自定义验证器 .....                                                  | 266 | 12.2.2 建立条件组合对象 .....               | 293        |
| 10.5.3 声明自定义验证器 .....                                                  | 268 | 12.2.3 实现数据层 .....                  | 294        |
| 10.5.4 引用自定义验证器 .....                                                  | 269 | 12.3 使用 Struts 2 来实现表现层 .....       | 302        |
| 10.6 引用验证器返回的错误信息 .....                                                | 270 | 12.3.1 将一组相关的 Action<br>类合成一个 ..... | 302        |
| 10.6.1 引用验证器返回的错误<br>信息 .....                                          | 270 | 12.3.2 显示全部用户 .....                 | 304        |
|                                                                        |     | 12.3.3 添加用户 .....                   | 306        |
|                                                                        |     | 12.3.4 修改用户 .....                   | 310        |
|                                                                        |     | 12.3.5 删除用户 .....                   | 314        |
|                                                                        |     | 12.3.6 按条件查询 .....                  | 316        |
|                                                                        |     | 12.3.7 小结 .....                     | 319        |
|                                                                        |     | 12.4 加入国际化信息 .....                  | 320        |
|                                                                        |     | 12.4.1 回顾国际化信息 .....                | 320        |
|                                                                        |     | 12.4.2 准备国际化信息文件 .....              | 321        |

|                                        |     |                                                        |     |
|----------------------------------------|-----|--------------------------------------------------------|-----|
| 12.4.3 让页面来使用国际化<br>信息.....            | 323 | 第 14 章 对 Ajax 的支持.....                                 | 355 |
| 12.4.4 切换语言.....                       | 324 | 14.1 Ajax 基础.....                                      | 356 |
| 12.5 加入验证.....                         | 325 | 14.1.1 Ajax 概述.....                                    | 356 |
| 12.5.1 回顾验证框架.....                     | 325 | 14.1.2 XMLHttpRequest 对象 ..                            | 356 |
| 12.5.2 为添加用户页面添加<br>验证.....            | 326 | 14.1.3 Ajax 示例.....                                    | 357 |
| 12.5.3 为查询用户页面添加<br>验证.....            | 327 | 14.2 Struts 2 对 Ajax 的支持 .....                         | 360 |
| 12.5.4 在验证中引用国际化<br>信息.....            | 329 | 14.2.1 概述 .....                                        | 360 |
| 12.6 加入拦截器.....                        | 330 | 14.2.2 名为 Stream 的 Result....                          | 361 |
| 12.6.1 回顾拦截器.....                      | 330 | 14.3 Ajax JSP tags.....                                | 363 |
| 12.6.2 编写自定义拦截器.....                   | 331 | 14.3.1 概述 .....                                        | 363 |
| 12.6.3 配置自定义拦截器.....                   | 332 | 14.3.2 head 标签.....                                    | 363 |
| 12.6.4 使用自定义拦截器的<br>效果.....            | 333 | 14.3.3 div 标签 .....                                    | 364 |
| 第 13 章 文件上传下载 .....                    | 335 | 14.3.4 a 标签和 submit 标签....                             | 368 |
| 13.1 文件上传.....                         | 336 | 14.3.5 bind 标签 .....                                   | 372 |
| 13.1.1 概述.....                         | 336 | 14.3.6 datetimepicker 标签.....                          | 375 |
| 13.1.2 使用 Struts 2 实现文件<br>上传.....     | 336 | 14.3.7 autocompleter 标签 .....                          | 377 |
| 13.1.3 获取文件的更多信息.....                  | 340 | 14.3.8 textarea 标签.....                                | 385 |
| 13.1.4 限制文件的大小及类型.....                 | 342 | 14.3.9 tabbedpanel 标签 .....                            | 386 |
| 13.1.5 上传超大的文件.....                    | 344 | 14.3.10 tree 和 treenode 标签....                         | 389 |
| 13.1.6 在一个表单中上传多个<br>文件.....           | 345 | 14.4 使用 JSON 插件.....                                   | 394 |
| 13.2 文件下载.....                         | 347 | 14.4.1 JSON 基础.....                                    | 394 |
| 13.2.1 概述.....                         | 347 | 14.4.2 JSON 插件.....                                    | 394 |
| 13.2.2 stream 结果类型 .....               | 347 | 14.4.3 使用 JSON 插件来实现<br>Ajax.....                      | 395 |
| 13.2.3 实现文件下载的<br>Action.....          | 348 | 第 15 章 整合 Spring .....                                 | 399 |
| 13.2.4 在 struts.xml 中配置<br>Action..... | 348 | 15.1 概述 .....                                          | 400 |
| 13.2.5 制作下载页面.....                     | 349 | 15.1.1 应用程序的分层 .....                                   | 400 |
| 13.2.6 难道只能在配置文件中写<br>死文件名吗? .....     | 350 | 15.1.2 使用 Spring 管理对象间<br>的依赖关系 .....                  | 402 |
| 13.2.7 处理中文文件名.....                    | 352 | 15.2 整合 Spring 与 Struts 2 .....                        | 402 |
|                                        |     | 15.2.1 概述 .....                                        | 402 |
|                                        |     | 15.2.2 复制到 jar 包 .....                                 | 402 |
|                                        |     | 15.2.3 改写 SampleAction .....                           | 402 |
|                                        |     | 15.2.4 编写 Spring 的配置文件<br>applicationContext.xml... .. | 403 |
|                                        |     | 15.2.5 在 web.xml 中引用 Spring<br>配置文件 .....              | 404 |
|                                        |     | 15.2.6 修改 struts.xml .....                             | 405 |

|                                              |     |                                        |     |
|----------------------------------------------|-----|----------------------------------------|-----|
| 15.3 自动装配 .....                              | 406 | 第 19 章 进度条 .....                       | 441 |
| 第 16 章 单元测试 .....                            | 409 | 19.1 使用“进度条”告知用户<br>进度 .....           | 442 |
| 16.1 概述 .....                                | 410 | 19.2 使用 execAndWait 拦截器模拟<br>进度条 ..... | 442 |
| 16.2 测试动作类 Action .....                      | 410 | 19.2.1 模拟长时间运行的<br>Action .....        | 442 |
| 16.2.1 测试动作类 Action 入门 .....                 | 410 | 19.2.2 配置 Action .....                 | 443 |
| 16.2.2 使用 JUnit 测试动作类<br>Action .....        | 412 | 19.2.3 实现“进度条”页面 .....                 | 443 |
| 16.3 测试完整的 Struts 2 运行流程 .....               | 414 | 19.2.4 实现完成页面 .....                    | 444 |
| 16.3.1 概述 .....                              | 414 | 19.2.5 运行测试 .....                      | 444 |
| 16.3.2 准备测试环境内 .....                         | 414 | 19.2.6 图形化的进度条 .....                   | 445 |
| 16.3.3 重写测试用例 .....                          | 416 |                                        |     |
| 16.3.4 运行测试用例 .....                          | 417 | 第 20 章 整合 SiteMesh .....               | 449 |
| 第 17 章 防止重复提交 .....                          | 419 | 20.1 调整应用的风格 .....                     | 450 |
| 17.1 什么是重复提交 .....                           | 420 | 20.2 单独使用 SiteMesh .....               | 451 |
| 17.2 使用 <s:token/> 标签 .....                  | 422 | 20.2.1 SiteMesh 下载 .....               | 451 |
| 17.2.1 使用 <s:token/> 标签入门 .....              | 422 | 20.2.2 复制资源 .....                      | 451 |
| 17.2.2 <s:token/> 的原理 .....                  | 424 | 20.2.3 引用过滤器 .....                     | 452 |
| 17.3 更强大的 tokenSession 拦截器 .....             | 425 | 20.2.4 定义模板页面 .....                    | 452 |
| 第 18 章 结合 JFreeChart .....                   | 427 | 20.2.5 定义被装饰页面 .....                   | 453 |
| 18.1 JFreeChart 使用 .....                     | 428 | 20.2.6 合成模板页面与被装饰<br>页面 .....          | 453 |
| 18.1.1 概述 .....                              | 428 | 20.3 整合 Struts 2 与 SiteMesh .....      | 454 |
| 18.1.2 JFreeChart 下载 .....                   | 428 | 20.3.1 准备资源 .....                      | 455 |
| 18.1.3 使用 JFreeChart 画出<br>饼图 .....          | 429 | 20.3.2 引用过滤器 .....                     | 455 |
| 18.1.4 使用 JFreeChart 画出柱<br>状图 .....         | 431 | 20.3.3 结合模板页面和被装饰<br>页面 .....          | 456 |
| 18.1.5 小结 .....                              | 433 | 第 21 章 零配置 .....                       | 459 |
| 18.2 Struts 2 插件概述 .....                     | 434 | 21.1 概述 .....                          | 460 |
| 18.2.1 Struts 2 插件的存在方式 .....                | 434 | 21.2 约定大于配置 .....                      | 460 |
| 18.2.2 Struts 2 插件的加载顺序<br>和作用 .....         | 435 | 21.2.1 约定大于配置的<br>HelloWorld .....     | 460 |
| 18.3 Struts 2 结合 JFreeChart .....            | 436 | 21.2.2 常见的约定 .....                     | 462 |
| 18.3.1 构建结合环境 .....                          | 436 | 21.2.3 示例常见的约定 .....                   | 464 |
| 18.3.2 Struts 2 结合 JFreeChart<br>来画饼图 .....  | 436 | 21.3 通过注解来实现零配置 .....                  | 465 |
| 18.3.3 Struts 2 结合 JFreeChart<br>来画柱状图 ..... | 438 | 21.3.1 概述 .....                        | 465 |
|                                              |     | 21.3.2 HelloWorld .....                | 465 |

|        |                                                 |     |            |                                      |     |
|--------|-------------------------------------------------|-----|------------|--------------------------------------|-----|
| 21.3.3 | @Action 和@Actions .....                         | 466 | A.1.10     | default-action-ref 元素 ..             | 477 |
| 21.3.4 | @Result 和@Results .....                         | 468 | A.1.11     | default-interceptor-ref<br>元素 .....  | 478 |
| 21.3.5 | @Namespace.....                                 | 469 | A.1.12     | default-class-ref 元素 ....            | 478 |
| 21.3.6 | @InterceptorRef 和<br>@InterceptorRefs .....     | 470 | A.1.13     | exception-mapping<br>元素 .....        | 479 |
| 21.3.7 | @ExceptionMapping 和<br>@ExceptionMappings ..... | 470 | A.1.14     | global-exception-mapping<br>元素 ..... | 479 |
| 21.3.8 | @ParentPackage .....                            | 471 | A.1.15     | global-results 元素 .....              | 479 |
| 21.3.9 | 用注解来替代 struts.xml                               | 471 | A.1.16     | interceptor 元素 .....                 | 479 |
| 附录 A   | Struts 2 的配置 .....                              | 473 | A.1.17     | interceptor-ref 元素 .....             | 480 |
| A.1    | struts.xml 的配置 .....                            | 474 | A.1.18     | interceptor - stack 元素..             | 481 |
| A.1.1  | 概述 .....                                        | 474 | A.1.19     | interceptors 元素 .....                | 481 |
| A.1.2  | DOCTYPE 和根元素 .....                              | 474 | A.1.20     | param 元素 .....                       | 482 |
| A.1.3  | package 元素 .....                                | 474 | A.1.21     | result 元素 .....                      | 482 |
| A.1.4  | action 元素 .....                                 | 475 | A.1.22     | result-type 元素 .....                 | 483 |
| A.1.5  | include 元素 .....                                | 475 | A.1.23     | result-types 元素 .....                | 483 |
| A.1.6  | bean 元素 .....                                   | 476 | A.2        | struts.properties 的配置 .....          | 483 |
| A.1.7  | constant 元素 .....                               | 476 | A.2.1      | 概述 .....                             | 483 |
| A.1.8  | unknown-handler-stack<br>元素 .....               | 477 | A.2.2      | 具体的参数描述 .....                        | 484 |
| A.1.9  | unknown-handler-ref<br>元素 .....                 | 477 | 参考文献 ..... |                                      | 488 |

## 第 1 章

# Struts 2 概述

## 1.1 MVC 模式

### 1.1.1 MVC 模式简介

MVC 是一种架构型模式，它本身并不引入新的功能，只是用来指导我们改善应用程序的架构，使得应用的模型和视图相分离，从而得到更好的开发和维护效率。

在 MVC 模式中，应用程序被划分成模型、视图和控制器三个部分。其中，模型部分包含了应用程序的业务逻辑和业务数据；视图部分封装了应用程序的输出形式，也就是通常所说的页面或者页面；而控制器部分负责协调模型和视图，根据用户请求来选择要调用哪个模型来处理业务，以及最终由哪个视图为用户做出应答。

MVC 模式的这三个部分的职责非常明确，而且相互分离，因此每个部分都可以独立地改变而不影响其他部分，从而大大提高应用的灵活性和重用性。

### 1.1.2 MVC 模式基础

#### 1. 模型、视图和控制器各个部分的作用

- 模型（Model）：负责封装应用的状态，并实现应用的功能。通常又分为数据模型和业务逻辑模型。数据模型用来存放业务数据，比如订单信息、用户信息等；而业务逻辑模型包含应用的业务操作，比如订单的添加或者修改等。
- 视图（View）：用来将模型的内容展现给用户，用户可以通过视图来请求模型进行更新。视图从模型获得要展示的数据，然后用自己的方式展现给用户，相当于提供页面来与用户进行人机交互；用户在页面上操作或者填写完成后，会单击提交按钮或是以其他触发事件的方式，来向控制器发出请求。
- 控制器（Controller）：用来控制应用程序的流程和处理视图所发出的请求。当控制器接收到用户的请求后，会将用户的数据和模型的更新相映射，也就是调用模型来实现用户请求的功能；然后控制器会选择用于响应的视图，把模型更新后的数据展示给用户。

#### 2. 模型和视图的关系

在 MVC 中，模型和视图是分离的，通常视图里面不会有任何逻辑实现；而模型也是不依赖于视图的，同一个模型可能会有很多种不同的展示方式，也就是同一个模型可以对应多种不同的视图。

这种例子比比皆是，例如，在 Windows 操作系统上浏览文件夹时，文件夹中的数据并没有变化，但是展示方式就有多种，比如大图标、小图标、详细信息等多种展示方式。以 tomcat7 文件夹中的文件为例，按照不同的展示方式，如图 1-1 所示。

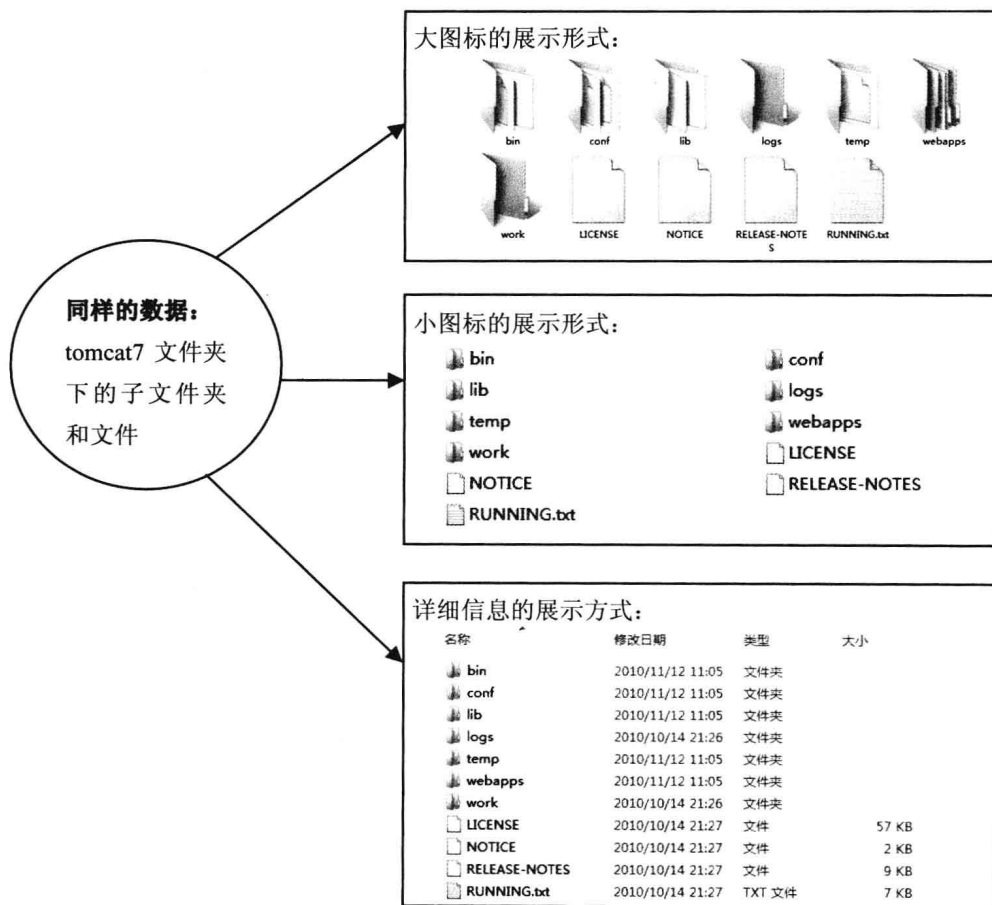


图 1-1 一个模型对应多个视图

模型负责输出的内容，而视图负责输出的形式，模型不依赖于视图，模型与视图是解耦的。因此在修改视图，也就是显示方式的时候，不必关心模型，而只需要直接修改视图的展示方式即可。

### 3. MVC 的组件关系图

MVC 的组件关系如图 1-2 所示。

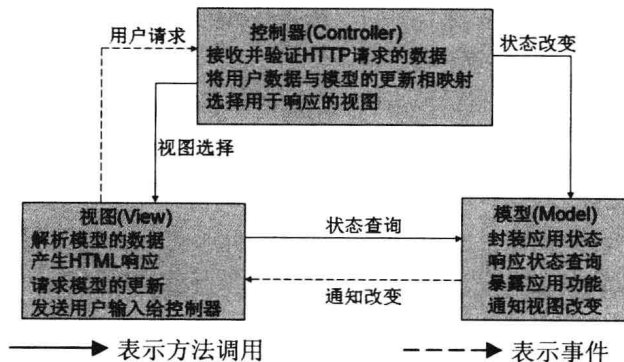


图 1-2 MVC 组件关系图

MVC 的组件关系图描述了模型、视图和控制器这三个部分的交互关系，下面按照交互顺序来详细描述一下它们的交互关系：

(1) 首先是展示视图给用户，用户在这个视图上进行操作，并填写一些业务数据。

(2) 然后用户会单击提交按钮，发出请求。

(3) 视图发出的用户请求会到达控制器，在请求中包含了想要完成什么样的业务功能以及相关的数据。

(4) 控制器处理用户请求，会把请求中的数据进行封装，然后选择并调用合适的模型，请求模型进行状态更新，然后选择接下来要展示给用户的视图。

(5) 模型处理用户请求的业务功能，同时进行模型状态的维护和更新。

(6) 当模型状态发生改变的时候，模型会通知相应的视图，告诉视图它的状态发生了改变。

(7) 视图接到模型的通知后，会向模型进行状态查询，获取需要展示的数据，然后按照视图本身的展示方式，把这些数据展示出来。

接下来就是等待用户下一次操作，再次从头轮回了。

### 1.1.3 MVC 模式作用

在早期开发中，有一些程序员没有认识到 MVC 模式带来的好处，在开发的时候不遵守 MVC 模式。这样做的结果就是程序结构划分不明确，各个部分功能混乱，在业务功能发生变更时，无论是业务逻辑修改还是显示形式修改，都要修改很多的类，“牵一发而动全身”，导致软件的开发和维护效率低下，错误百出。

而遵循 MVC 模式来开发系统，就会极大地避免上述问题的出现。MVC 模式的核心手段是解耦，MVC 模式通过仔细的划分功能，把整个应用程序划分成模型、视图、控制器三个部分，然后严密控制三个部分之间的通信，从而得到一个结构清晰、功能分布合理、可重用、可扩展、可维护的应用程序。

因此，使用 MVC 模式，可以获得以下好处：

- 低耦合性：在 MVC 模式中，模型和视图是解耦的，模型不会依赖于视图，而视图也仅仅是从模型中获取需要展示的数据，并不会与模型的逻辑处理相关联。
- 更低的开发成本：由于 MVC 模式清楚地划分了各部分的职责，就可以让程序员各司其职，Java 程序员只关心业务逻辑的实现，也就是模型部分；而页面程序员只关心页面展示，也就是视图部分即可。
- 更好的可维护性：MVC 模式划分出明晰的模型和视图部分，并使其解耦，在软件需求发生变更的时候，就可以各自独立地改变而不会相互影响，使得程序更容易维护和扩展。

### 1.1.4 Model2 模型中的 MVC

在 Java 的 Web 开发中，通常把 Servlet+JSP+JavaBean 的模型称为 Model2 模型，这是一个完全遵循 MVC 模式的模型，基本划分如下：

- **JavaBean** 作为模型，既可以作为数据模型来封装业务数据，又可以作为业务逻辑模型来包含应用的业务操作。其中，数据模型用来存储或传递业务数据，而业务逻辑模型接收到控制器传过来的模型更新请求后，执行特定的业务逻辑处理，然后返回相应的执行结果。
- **JSP** 作为表现层，负责提供页面来为用户展示数据，提供相应的表单 (Form) 用于用户的请求，并在适当的时候（比如用户单击提交的按钮），向控制器发出用户请求来请求模型进行更新。
- **Servlet** 作为控制器，用来接收用户提交的请求，然后获取请求中的数据，将其转换为业务模型需要的数据模型，然后调用业务模型相应的业务方法，请求模型进行更新，同时根据业务执行结果来选择要返回的视图，也就是选择下一个页面。
- Model2 实现 MVC 的基本结构图如图 1-3 所示。

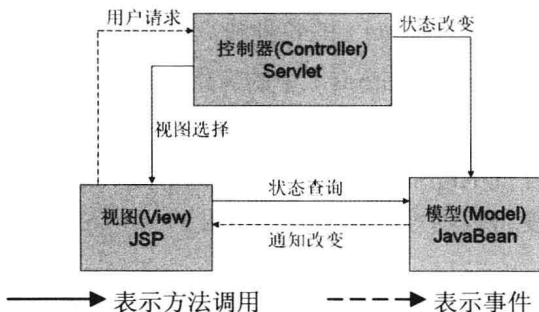


图 1-3 Model2 中的 MVC

那么，在用户发出一个请求后，Servlet+JSP+JavaBean 模型基本的响应顺序是怎样的呢？

当用户发出一个请求后，这个请求会被控制器 Servlet 接收到；Servlet 将请求的数据转换成数据模型 JavaBean，然后调用业务逻辑模型 JavaBean 的方法，并将业务逻辑模型返回的结果放到合适的地方，比如请求的属性里；最后，根据业务逻辑模型的返回结果，由控制器来选择合适的视图 (JSP)，由视图把数据展现给用户。

## 1.2 Struts 2 基础

### 1.2.1 Struts 1 时代

随着时间的推移，Servlet+JSP+JavaBean 模型暴露出它的缺点：

- **流程凌乱**：Servlet 在完成对用户请求的处理后，下一个页面是谁？如何跳转