

21世纪

软件工程师培养优秀教材



1CD



Struts

网络编程 例学与实践



闻涛 编著



清华大学出版社

Struts 网络编程例学与实践

闻 涛 编著

清华大学出版社

北 京

内 容 简 介

本书是一本 Struts 的技术应用教程，目的是为准备了解和使用 Struts 的技术人员提供指导和帮助。

本书共分为 12 章，内容以实用为主，原理和实例相辅相成，由浅入深，循序渐进。前 7 章是 Struts 的基础，其中有 Struts 基本概念、Struts 的 ActionForm、Struts 标签和 Struts 的 Action 等内容；第 8~11 章是 Struts 比较高级的内容，包括 Struts 自定义标签、Struts 校验机制、Struts 国际化方法和 Struts 的 Tiles 组件；第 12 章是一个完整实例，结合 Eclipse、Ant 等内容，搭建了一个真实的开发环境和实际的应用。

本书内容基础实用，既可以作为广大工程技术人员的参考，也适合大中院校的学生作为入门教程。

本书的示例都是精心挑选的，所有代码都可以从所附光盘中获得。

版权所有，翻印必究。举报电话：010-62782989 13501256678 13801310933

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。

本书防伪标签采用特殊防伪技术，用户可通过在图案表面涂抹清水，图案消失，水干后图案复现；或将表面膜揭下，放在白纸上用彩笔涂抹，图案在白纸上再现的方法识别真伪。

图书在版编目 (CIP) 数据

Struts 网络编程例学与实践/闻涛主编. —北京:

清华大学出版社, 2006.3

ISBN 7-302-12478-7

I. S… II. 闻… III. 软件工具—程序设计

IV. TP311.56

中国版本图书馆 CIP 数据核字 (2006) 第 006270 号

出 版 者: 清华大学出版社

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

社 总 机: 010-62770175

地 址: 北京清华大学学研大厦

邮 编: 100084

客户服务: 010-82896445

组稿编辑: 夏非彼

文稿编辑: 刘秀青

封面设计: 林陶

版式设计: 科海

印 刷 者: 北京科普瑞印刷有限责任公司

发 行 者: 新华书店总店北京发行所

开 本: 正 16 印张: 16.25 字数: 395 千字

版 次: 2006 年 4 月第 1 版 2006 年 4 月第 1 次印刷

书 号: ISBN 7-302-12478-7/TP · 8001

印 数: 0001~4000

定 价: 32.00 元 (ICD)

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：(010) 82896445

前 言

Struts 目前已经成为 Web 开发的主流框架之一, 并且正有越来越多的开发组织在进行着 Struts 的应用开发实践。Struts 的参考手册和指导书籍也有很多, 但是大多数都理论重于实际, 针对开发中的具体问题和细节设计不够深入, 这样就造成看完书后仍无法上手的尴尬局面。本书针对这一问题, 在讲解理论的基础上, 通过大量生动的例子来帮助读者理解, 并最大程度地保证读者在阅读完本书后能够实际上手进行开发工作。

本书主要是为从事 Web 开发的广大工程技术人员所写。作为 Web 开发的主流框架之一, Struts 是每一个程序员应该了解并掌握的, 特别是如果您希望从事这方面的工作。另外, 本书也适合各院校对 Web 开发有一定了解的学生用于学习。

本书基本内容组织如下:

- 第 1 章简要回顾了 Struts 的历史;
- 第 2 章介绍了 Struts 的基本概念和一个简短的小例子;
- 第 3 章讲解了 Struts 的 ActionForm;
- 第 4~6 章详细讲解 Struts 的几个重要标签库;
- 第 7 章讲解了 Struts 的 Action;
- 第 8 章是关于自定义标签;
- 第 9 章论述了 Struts 的校验机制;
- 第 10 章说明了 Struts 的国际化方法;
- 第 11 章讲解了 Struts 的 Tiles 组件;
- 第 12 章给出一个基于 Struts 的完整应用系统。

本书前 7 章是 Struts 的基础内容。如果您对 Struts 比较陌生, 最好从头开始阅读; 第 8~11 章是 Struts 的高级内容, 各章比较独立, 如果您对 Struts 已经比较熟悉, 可以挑选您需要和感兴趣的部分单独进行阅读; 如果您已经看过不少 Struts 方面的书籍, 只是没有实际做过开发, 不妨看看第 12 章的内容, 或许它对您是一个惊喜。

本书的所有源代码都可以在随书光盘中找到, 方便各位读者的学习和使用。

没有各位朋友的关心和帮助, 这本书是不可能与广大读者见面的; 在这里, 衷心感谢各位朋友在此书的写作和出版过程中给予的帮助!

作 者

2006 年 2 月

目 录

第 1 章 Web 应用和 Struts	1
1.1 Web 应用技术的发展	1
1.1.1 静态页面	1
1.1.2 动态页面	1
1.2 Struts 介绍	3
第 2 章 Struts 基本原理	4
2.1 MVC 模式	4
2.2 Struts 基本概念	5
2.2.1 体系结构	5
2.2.2 配置文件 struts-config.xml	6
2.3 Struts 工作流程	7
2.4 一个最简单的例子: Hello	8
2.4.1 创建本地工作区和 Web 环境	8
2.4.2 配置 Struts 环境	10
2.4.3 实例代码	11
第 3 章 处理外部输入的利器——ActionForm	14
3.1 什么是 ActionForm	14
3.2 ActionForm 实例	14
3.3 ActionForm 中两个重要的方法	17
3.3.1 validate()	17
3.3.2 reset()	17
3.4 ActionForm 的作用范围	17
3.5 动态 ActionForm	18
3.5.1 后端映射 (mapbacked) ActionForm	18
3.5.2 动态 ActionForm (DynaActionForm)	19
第 4 章 Struts 的标签库之一: <html>标签	21
4.1 <html:html>标签	21
4.2 <html:form>标签	23
4.3 <html:text>标签	25
4.4 <html:submit>标签	27
4.5 <html:button>标签	29

4.6	<html:textarea>标签	30
4.7	<html:checkbox>标签	31
4.8	<html:multibox>标签	33
4.9	<html:radio>标签	35
4.10	<html:select>标签和<html:option>标签	37
4.11	<html:file>标签	38
4.12	<html:image>标签	39
4.13	<html:hidden>标签	42
4.14	<html:link>标签	42
4.15	<html:img>标签	43
4.16	<html:errors>标签	44
4.17	综合实例	46
第 5 章	Struts 的标签库之二: <bean>标签	51
5.1	<bean:struts>标签	51
5.2	<bean:cookie>标签	52
5.3	<bean:define>标签	54
5.4	<bean:header>标签	55
5.5	<bean:parameter>标签	57
5.6	<bean:write>标签	58
5.7	<bean:message>标签	60
5.8	<bean:include>标签	62
5.9	<bean:page>标签	64
5.10	<bean:resource>标签	66
5.11	<bean:size>标签	67
第 6 章	Struts 的标签库之三: <logic>标签	70
6.1	<logic:present>标签和<logic:notPresent>标签	70
6.2	<logic:iterate>标签	74
6.3	<logic:equal>标签和<logic:notEqual>标签	76
6.4	<logic:greaterEqual>标签和<logic:greaterThan>标签	78
6.5	<logic:lessEqual>标签和<logic:lessThan>标签	79
6.6	<logic:match>标签和<logic:notMatch>标签	80
6.7	<logic:empty>标签和<logic:notEmpty>标签	84
6.8	<logic:forward>标签	86
6.9	<logic:redirect>标签	87
6.10	<logic:messagePresent>和<logic:messageNotPresent>标签	90

第 7 章 Struts 世界的控制者——Action	93
7.1 Struts 的 Action	93
7.2 使用 DispatchAction	96
7.3 使用 LookupDispatchAction	99
7.4 使用 ForwardAction	103
7.5 使用 IncludeAction	108
7.6 使用 SwitchAction	110
第 8 章 标签库的扩展：自定义标签	116
8.1 什么是自定义标签	116
8.1.1 声明 tag 标签所在的 tag 库	117
8.1.2 实现 tag 标签	117
8.2 使用自定义标签 currentDatetime	117
8.2.1 创建 currentDatetime 标签的 TLD 文件	119
8.2.2 创建 currentDatetime 标签的 Java 文件	120
8.3 自定义标签应用实例	124
8.3.1 带属性的标签	124
8.3.2 自定义的 Iterator 标签	127
8.3.3 自定义变量生成标签	131
8.3.4 创建脚本变量的自定义标签	134
8.4 小结	138
第 9 章 Struts 的校验机制	139
9.1 传统的 JavaScript 校验	140
9.2 Struts 的一般校验方法	146
9.2.1 ActionErrors 对象	146
9.2.2 ActionForm 和 Action 中数据校验实例	148
9.3 Struts Validator 框架	155
9.3.1 Validator 的配置	156
9.3.2 validator-rules.xml 文件	157
9.3.3 validation.xml 文件	160
9.3.4 ActionForm 的使用	162
9.3.5 Validator 框架应用实例	164
9.3.6 用户自定义校验规则	168
9.3.7 Validator 框架与 JavaScript 校验	171
9.3.8 Validator 框架对于国际化的支持	172
9.4 小结	173

第 10 章 资源文件.....	174
10.1 创建和使用资源文件.....	174
10.2 在 Action 类中使用资源文件.....	175
10.3 多资源文件和国际化.....	178
10.4 资源文件中的编码问题.....	179
第 11 章 装饰大师: Tiles	181
11.1 准备工作	181
11.2 创建一个经典页面布局.....	183
11.3 使用<tiles:definition>标签.....	186
11.4 Definition 的扩展	188
11.5 通过 Action 调用 definition	190
第 12 章 Struts 应用实例: 在线用户注册系统.....	192
12.1 目标系统	192
12.2 系统展示	193
12.3 利用 Eclipse 开发 Web 应用程序.....	195
12.3.1 新建 Web 项目	195
12.3.2 建立源程序文件夹.....	199
12.3.3 添加外部 jar 包的引用.....	202
12.3.4 为项目添加文件夹.....	208
12.3.5 项目结构说明.....	209
12.3.6 项目的部署.....	210
12.3.7 小结	211
12.4 在线用户注册系统开发实战.....	212
12.4.1 数据库设计.....	212
12.4.2 系统框架功能的实现.....	212
12.4.3 第一个页面——系统登录页面的实现	218
12.4.4 浏览用户列表页面的实现	224
12.4.5 新增用户页面的实现.....	229
12.4.6 修改和删除用户信息.....	236
12.4.7 用户权限控制的功能.....	239
12.4.8 小结	242
12.5 在 Eclipse 中利用 Ant 脚本部署应用程序	242
12.5.1 配置 Eclipse 中 Ant 脚本功能的相关参数.....	242
12.5.2 在 Eclipse 中编写 Ant 脚本	244
12.5.3 运行 Ant 脚本.....	247
12.5.4 Ant 脚本视图.....	250
12.5.5 小结	252

第 1 章

Web 应用和 Struts

1.1 Web 应用技术的发展

Web 从诞生至今，其影响和应用已经无处不在。由于人们对 Web 的需求不断增加，Web 技术有了深入的发展。从最初的 HTML 语言，到 CGI、API，再到如今的 ASP、JSP、Struts，Web 应用技术的不断发展为各式各样日益复杂的 Web 应用提供了前提和保障。下面我们首先来简短地回顾一下 Web 应用技术发展的历史。

1.1.1 静态页面

静态页面是指最简单的网页，一般不包含动画和程序，不能进行交互，不涉及数据库。静态页面技术的使用方式是：客户端向服务器发送请求，服务器收到请求后将被请求的文件发往客户端。客户端收到的文件副本与服务器上的文件完全相同。在正常情况下，Web 页面总是保持原样不变，多个客户对同一静态页面的请求会得到相同的结果。

静态页面出现在 Web 应用的早期，功能比较单一，早已远远不能满足当今 Web 应用的要求了，目前已经很少有纯静态的 Web 应用了。

1.1.2 动态页面

动态页面是指在被请求时才创建的 Web 页面，并且页面内容依赖于特定的规则。当服务器接收到客户端的请求时，服务器端的应用程序会根据客户的输入和特定的规则，动态地生成 Web 页面，即动态页面。动态页面一般具有下面两个特性：

- ❶ 交互性：网页会根据用户的动作进行动态改变和响应。
- ❷ 多样性：页面内容随时间和用户的不同而变化。

下面分别来介绍动态页面的几种相关技术。

1. CGI (Common Gateway Interface)

CGI 是在 Web 上创建动态网页的一种方案。它采用接口方式，允许通过外部程序来接收请求，而且外部程序可以用各种语言来编写，比如 C、C++、Perl 等。CGI 的出现让 Web 从静态走向动态，然而，由于其自身的一些缺点，限制了其进一步的发展。比如：

- ❶ 运行效率低、速度慢：每生成一个动态页面，系统就必须启动一个新的进程以运行 CGI 程序，十分消耗系统资源，进而会影响速度。
- ❷ 不易编写：CGI 属于底层操作，远不及 ASP、JSP 和 PHP 容易。如果我们把 JSP 等比作高级语言，那么 CGI 就是汇编语言了。

2. API (Application Program Interface)

动态 Web 应用的另一个解决方案是专用于 Web 服务器的 API。不同的 Web 服务器提供的 API 不同，比如 Netscape 提供的是 NSAPI，Microsoft IIS 提供的是 ISAPI。API 方案的资源利用率高、执行速度快，它一般能比 CGI 快几倍，较好地解决了效率和速度的问题。

但是 API 方案的平台依赖性太强，应用程序完全受限于所使用的网站服务器平台，不能改变所使用的网站服务器。另外，应用程序直接在服务器的执行进程内被呼叫，使整个网站服务器都受到应用程序的影响。

3. Script (客户端脚本语言)

客户端脚本语言技术提供了另一种动态页面的实现方法。它是将代码发送到客户端，并在客户端运行。常见的 JavaScript 与 VBScript 都是属于这种技术。客户端脚本语言技术可以大大减少客户端与服务器端的交互，减轻服务器端的负担，提高效率，如今这种技术已经广泛应用。但客户端脚本语言技术本身比较简单，难以胜任较为复杂的应用，所以它更多的是作为一种辅助技术使用。另外，因为脚本语言是在客户端执行的代码，所以会造成一些安全隐患，再加上不同的浏览器及浏览器的不同版本对脚本语言的支持也不相同，在一定程度上也造成了脚本语言技术在开发、维护和应用上的困难。

4. ASP (Active Server Pages) 和 JSP (JavaServer Page)

ASP 全名 Active Server Pages，是一个 Web 服务器端的开发环境，利用它可以产生和执行动态的 Web 服务应用程序。ASP 采用脚本语言作为自己的开发语言，支持将 Html 脚本和可重用的 Active Server 组件结合在一起构建交互式网页。当服务器收到客户端的请求后，先在服务器端运行相应的脚本，然后再将处理结果发送到客户端。

ASP 功能丰富，使用难度小，目前在 Windows 平台上应用较为广泛；但是 ASP 是 Microsoft 开发的动态网页语言，因此只能执行于微软的服务器产品上，平台依赖性使它目前难以克服的障碍。

JSP 是 Sun 公司推出的 Web 开发语言。JSP 分离了内容的产生和显示，使用 JSP 标识或者小脚本来产生页面上的动态内容。产生内容的逻辑被封装在 JavaBeans 中。所有的脚本在服务器端执行，JSP 引擎在服务器端解释 JSP 标识，产生所请求的内容，并且将结果以 HTML 页面的形式发送回客户端浏览器。因为封装在 JavaBeans 中的业务逻辑提供了重用的可能，标准的 JSP 标识提供了各种功能，这大大简化了开发的难度。

JSP 能够支持复杂的 Web 应用，并且作为 Java 平台的一部分，JSP 拥有 Java 程序设计语言“一次编写，各处执行”的特点。目前 JSP 已经成为 Web 开发的主流技术。

5. Struts

Struts 不同于上面所讲的各种技术，它不是一种具体的 Web 开发技术。简单的说，Struts 为基于 Java 的 Web 应用程序开发提供了一个框架。基于这个框架，用户可以快速、高效地实现复杂的 Web 应用。

下一节将为您更详细地介绍有关 Struts 的情况。

1.2 Struts 介绍

Struts 最早是 Apache Jakarta 项目的组成部分。项目的创立者希望通过对该项目的研究，改进和提高 JavaServer Pages、Servlet、标签库以及面向对象的技术水平。Struts 这个名字来源于在建筑和旧式飞机中使用的支撑金属架，Struts 代表支撑各种建筑物的基础支架。这也是对 Struts 在开发 Web 应用程序中所扮演的角色的精彩描述。当建立一个物理建筑时，建筑工程师使用支柱为建筑的每一层提供支持，同样，软件工程师使用 Struts 为应用的每一层提供支持。

Struts 基础代码从 2000 年 5 月开始开发，2001 年 6 月发布 1.0 版本，2003 年 6 月发布 1.1 版，2004 年 7 月发布 1.2.1 版，目前可得到的最新版本是 1.2.7。

Struts 为 Web 应用提供了一个通用的架构，这样开发人员可以把精力集中在如何解决实际问题。同时，Struts 还能够在适当的地方提供扩展接口，使应用程序能扩展该框架来更好地适应使用者的实际需要。Struts 具有如下的特点：

- ❶ 基于开放源代码。
- ❷ 支持 MVC 架构。
- ❸ 支持丰富的标签库。
- ❹ 分离了显示层、业务逻辑和控制层。
- ❺ 提高了代码的可重用性、可维护性、可扩展性。

由此可见，对于大型的、复杂的 Web 应用来说，Struts 提供了一个优秀的架构。在下一章中，您将首先了解到 Struts 的一些具体内容，然后会看到一个简短的小例子。

第 2 章

Struts 基本原理

2.1 MVC 模式

JSP 技术通常应用的模式是 JB 模式，即 JSP+JavaBean 模式，如图 2-1 所示。

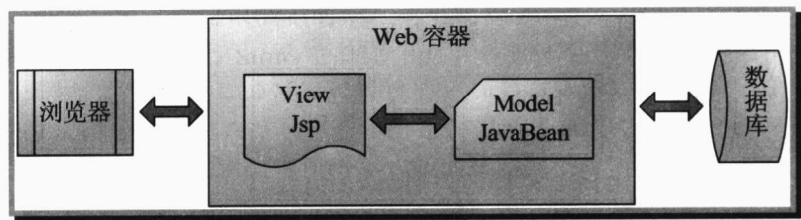


图 2-1 JB 结构图

这种模式分离了用户视图和业务逻辑：将前端的视图用 JSP 实现，后台的业务逻辑封装在 JavaBean 中。也就是使前端的开发人员和后台的业务逻辑实现人员各司其职，大大简化了开发工作，提高了可重用性和可维护性。

然而，随着 Web 应用程序复杂度的提高，这种模式变得越来越不适用。为了更好地解决应用中的问题，MVC 模式应运而生。MVC 模式即 Model+View+Control 模式，如图 2-2 所示。

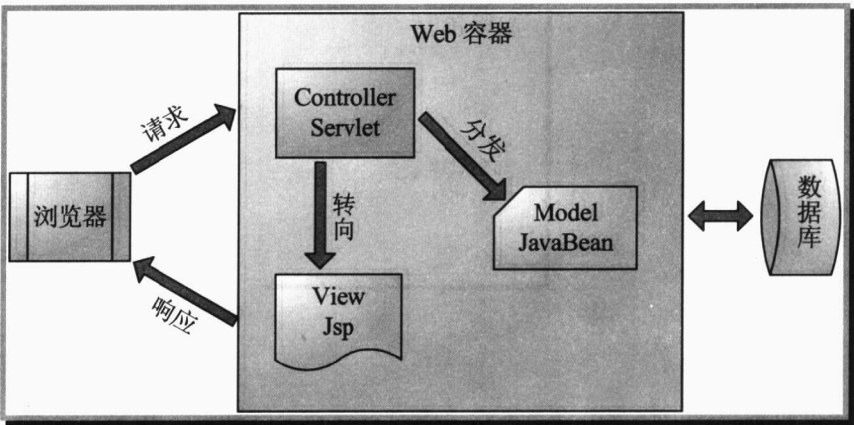


图 2-2 MVC 结构图

MVC 模式把一个系统分成 3 个相互协调的部分：

- ❶ 模型（Model）用于封装系统的状态，它表示一个应用系统的数据并且包含访问和管理该数据的逻辑。模型聚集了相关的数据和操作以提供服务。通过控制器访问模型服务，模型的状态会发生相应的变化，这些变化会通报给视图。
- ❷ 视图（View）是模型的表示，提供用户交互界面。当模型状态发生变化时，视图应该得到通知以便更新。视图还要将用户输入传递到控制器。
- ❸ 控制器（Controller）接受来自视图的请求，以修改模型的状态。控制器的任务是获取用户输入并交由模型执行，然后根据用户输入和模型执行结果选择相应视图。

MVC 模式将用户的请求和系统的响应封装在一起，交由 Controller 统一处理。用户视图 View 和模型 Model 使用 JSP 和 JavaBean 实现，这样用户请求和系统响应便从用户视图和业务逻辑中分离出来，使得应用程序结构更加清晰，控制更加容易，维护更加方便，可重用性也大大提高。

2.2 Struts 基本概念

2.2.1 体系结构

Struts 就是基于 MVC 模式设计的 Web 框架，因此适合于构建复杂的 Web 应用。Struts 结构如图 2-3 所示。

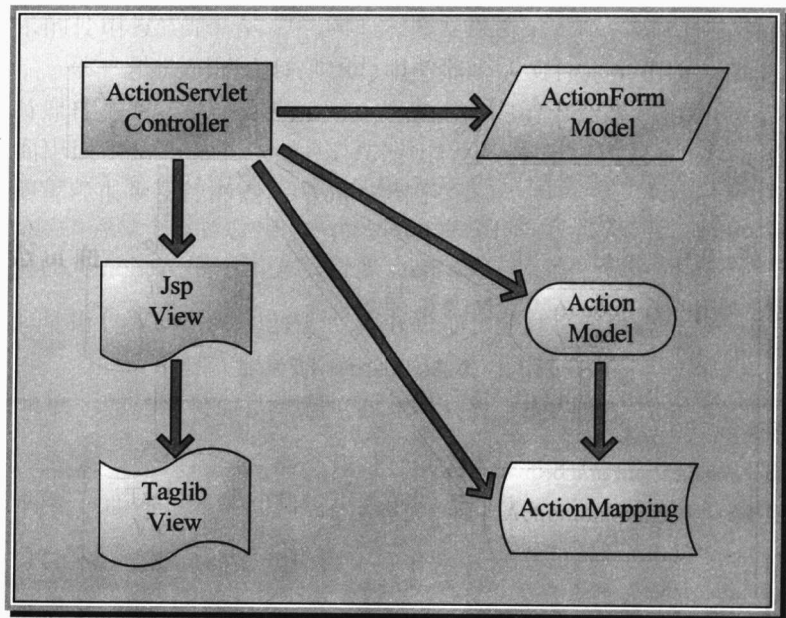


图 2-3 Struts 结构图

从图 2-3 可以看出，通过 Struts 框架的组件，可以将一个 Web 应用分解为三部分：用户视图（View）、控制逻辑（Controller）和业务逻辑（Model）。

Struts 框架相应的有三个主要部件：ActionServlet，控制程序的流转；Action 类，负责处理具体业务；Jsp 页面，显示用户视图。Struts 的中心控制器 ActionServlet 接受所有来自客户端的请求，并根据系统的配置文件 Struts-config.xml 将来自客户端的请求转发到相应的 Action 对象，所有的业务操作都在这些 Action 对象中进行。Action 对象处理完毕后，由 ActionServlet 转向到 JSP 页面，将处理结果返回给客户端。

另外，Struts 还提供了一组标签库（Bean Tags，HTML Tags，Logic Tags，Template Tags 等）。使用标签库可以极大地减少 JSP 端的 Java 代码，简化了创建用户界面的过程。

2.2.2 配置文件 struts-config.xml

struts-config.xml 是 Struts 的配置文件，主要包括下面两部分的内容：

1. ActionMapping 的配置

ActionMapping 的作用是进行流程控制，将请求 URI 映射到 Action 类，并且将 Action 类与 ActionForm bean 相关联。ActionServlet 在内部使用这些映射，并将控制转移到特定 Action 类的实例。所有 Action 类均返回一个 ActionForward 对象，其中包括响应转发的目标资源名称。例如：

```
<action-mappings>
<action path="/HelloAction"
        name="HelloActionForm"
        type="cn.struts.HelloAction"
        scope="request"
        validate="false">
    <forward name="success" path="/jsp/hello.jsp" />
</action>
</action-mappings>
```

上面的代码段定义了一个最简单的 ActionMapping，其中只包含了一个流转控制，就是位于<action></action>之间的部分。通常 ActionMapping 中会有若干这样的<action></action>对，它们定义了应用程序的所有流转控制。path 属性定义了引用 Action 类的相对路径；name 指定了相关联的 ActionFormbean。其他请见表 2-1。

表 2-1 ActionMapping 的属性

属性	描述
name	相关联的 ActionForm bean 的名称
Type	连接到本映射的 Action 类的全称（可有包名）
path	Action 类的相对路径
scope	ActionForm bean 的作用域（请求或会话）
input	输入表单的路径，指向 bean 发生输入错误时必须返回的控制
validate	如果设置为 true，则将调用 ActionForm bean 的 validate()方法来进行输入校验
prefix	用来匹配请求参数与 bean 属性的前缀
suffix	用来匹配请求参数与 bean 属性的后缀

(续表)

属性	描述
attribute	作用域名称
unknown	如果设置为 true, 操作将被作为所有没有定义的 ActionMapping 的 URI 的默认操作

上面代码中的 forward 元素用来指定要转向到的资源, 其属性见表 2-2。

表 2-2 forward 元素的属性

属性	描述
name	Action 类转向时所用的逻辑名
path	转向的目标路径
redirect	如果设置为 true, 则 ActionServlet 使用 sendRedirect()方法来转发资源

2. ActionForm Bean 的配置

ActionForm 用来保存请求的参数, 这些 bean 的属性名称与 Http 请求参数中的名称相对应, 控制器将请求参数传递到 ActionForm bean 的实例, 然后将这个实例传送到 Action 类。例如:

```
<form-bean name="HelloActionForm"
           type="cn.struts.HelloActionForm" />
</form-beans>
```

上面的代码段就定义了一个 ActionForm, 它的名字由 name 属性指定, 原则上命名没有限制, 但是通常将其名字与其类名保持一致, 便于记忆和使用; 它的 type 属性则指定了它的类名。

ActionForm 的属性见表 2-3。

表 2-3 ActionForm 的属性

属性	描述
name	在 ActionMapping 中的引用名
type	类的完全限定名

2.3 Struts 工作流程

启动 Struts 时, Controller 读入配置文件 struts-config.xml, 文件中定义了前端请求的 URL 与相应的 Action 类及使用的 Form 类的映射关系。Struts 的工作流程如图 2-4 所示。

- (1) 浏览器向服务器发出请求, Controller 首先获得这些请求。
- (2) Controller 根据配置文件 struts-config.xml 中的定义将这些请求发送到相应的 Action, 对应图中的 Model。
- (3) Action 完成相应的业务逻辑处理后, 将处理结果返回到 Controller。

(4) Controller 再根据配置文件 struts-config.xml 中的定义将处理结果显示到用户视图 View。

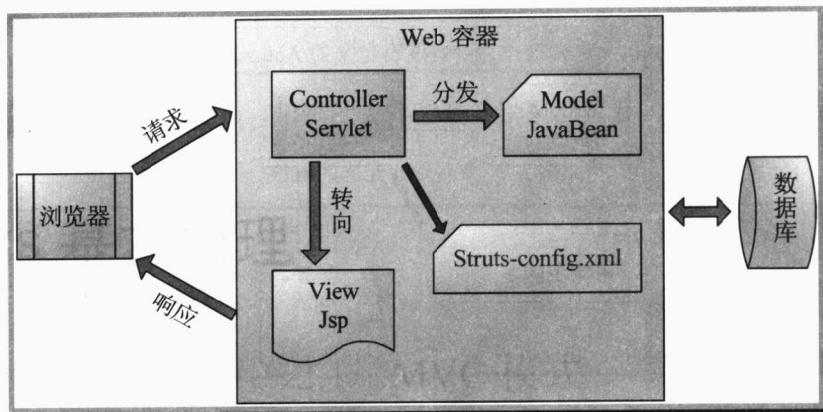


图 2-4 Struts 工作流程示意图

如果您对上述内容感觉不是很清晰的话，请不要着急。下一小节，我们将通过一个小而全的例子向您更加细致地展示 Struts 的配置和使用。您不妨看完下一小节再回来理解上面的内容。

2.4 一个最简单的例子：Hello

2.4.1 创建本地工作区和 Web 环境

第一步，您需要创建一个工作区目录，目录结构如图 2-5 所示。

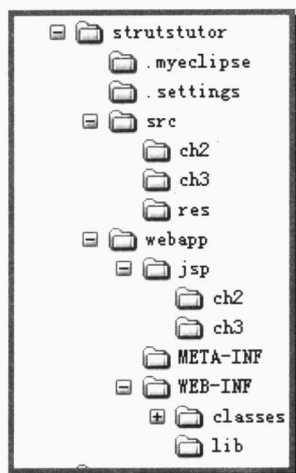


图 2-5 工作区目录结构

各目录的功能如下：

- ❶ src 目录：源程序目录，存放本实例用到的 java 类和资源文件。
- ❷ webapp 目录：Web 应用程序目录，以后要将这个目录部署到 Tomcat 下。
- ❸ webapp\jsp 目录：Web 应用程序用到的 Jsp 存放在这个目录下面。
- ❹ WEB-INF 目录：存放 web.xml、struts.xml 等系统文件。
- ❺ WEB-INF\lib 目录：存放 Web 应用程序用到的 jar 包。本实例需要把 Struts 压缩包中的 lib 目录下的文件全部复制过来。
- ❻ WEB-INF\classes 目录：编译好的 java 类放到这里。

在本书中，我们选用 Tomcat 作为 Web 服务器，Tomcat 的最新版本号是 4.1.31，到 <http://apache.justdn.org/jakarta/tomcat-4/v4.1.31/bin/jakarta-tomcat-4.1.31.exe> 可以下载安装该版本。有关 Web 服务器的安装和配置请参考有关书籍，这里我们只介绍跟 Struts 相关的 Tomcat 配置。

第二步，配置 Tomcat。目录结构如图 2-6 所示。

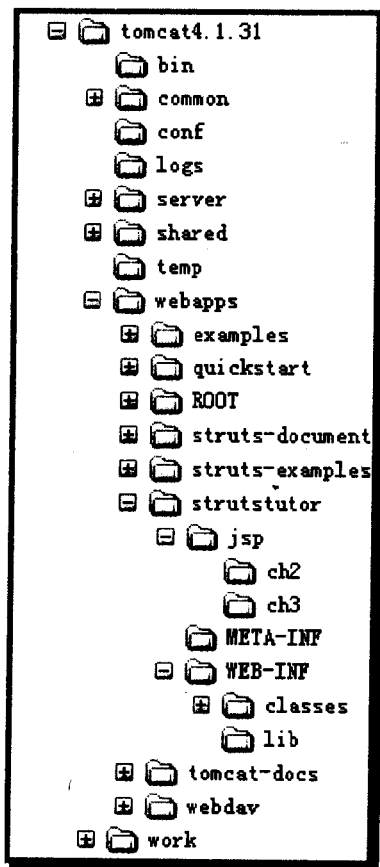


图 2-6 Tomcat 配置结构图

把本地工作区目录 webapp 下的内容直接复制到 Tomcat 目录中 webapps 中的 struttutor 目录下就可以了。