



实战 STRUTS

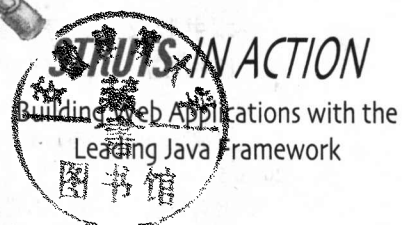
STRUTS IN ACTION

Building Web Applications with the
Leading Java Framework

(美) Ted Husted 等著
黄若波 程峰 程繁科 译



实战 STRUTS



(美) Ted Husted 等著
黄若波 程峰 程繁科 译



机械工业出版社
China Machine Press

本书是由 Struts 开发小组的核心成员编写的一本关于 Struts 开发的经典书籍。它一步步地由浅入深, 将 Struts 分解成相对独立的部分进行了讲解。本书除了介绍 Struts 以外, 同时还介绍了一些与 Struts 相关的组件, 合理地使用这些组件可以帮助程序员开发出更加健壮、更加专业、更加容易维护的应用程序。对于刚刚接触 Struts 的新手而言, 本书无异于是学习和掌握 Struts 的圣经。对于在 Struts 方面有不少经验的程序员而言, 本书也提供了 Struts 底层内容, 帮助读者更好地编写基于 Struts 的应用程序。

Ted Husted, et al: Struts in Action, Building Web Applications with the Leading Java Framework (ISBN 1-930110-50-2).

Original English edition published by Manning Publications Co., 209 Bruce Park Avenue, Greenwich, Connecticut 06830.

Copyright ©2003 by Manning Publications Co.

All rights reserved.

Simplified Chinese translation edition published by China Machine Press.

Copyright ©2005 by China Machine Press.

本书中文简体字版由 Manning 出版公司授权机械工业出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

版权所有, 侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号: 图字: 01-2005-2716

图书在版编目 (CIP) 数据

实战 Struts/ (美) 赫斯特 (Husted, T.) 等著; 黄若波等译. - 北京: 机械工业出版社, 2005.5

书名原文: Struts in Action: Building Web Applications with the Leading Java Framework
ISBN 7-111-16281-1

I. 实… II. ①赫…②黄… III. 软件工具—程序设计 IV. TP311.56

中国版本图书馆 CIP 数据核字 (2005) 第 025526 号

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑: 周莉梅 刘立卿

北京奔腾印刷厂印刷·新华书店北京发行所发行

2005 年 5 月第 1 版第 1 次印刷

787mm × 1092mm 1/16 · 30.25 印张

印数: 0 001-4 000 册

定价: 55.00 元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换
本社购书热线: (010) 68326294

序 言

本书是很多资深的 Struts 开发人员艰苦工作的结晶。Ted、Cedric、George 和 David 做得非常出色，讲述了 Struts 是如何工作的，以及如何在实际工作中使用 Struts。如果你是一个使用 Struts 的新手，则本书可以使你更快更容易地掌握 Struts 框架结构，并将它使用到自己的实际项目中去。而那些已经很了解 Struts 的程序员，通过阅读本书，还是可以发现一些新的内容。

我是在 20 世纪 90 年代末期开始对 Web 应用程序感兴趣的。当时我已经掌握了一门计算机语言（Java），它解决了高级编程中一个很重要的问题——程序员不需要考虑如何动态地释放已经被分配的内存。

在开始的时候，我所期望的仅仅是，通过我的努力使得构造基于 Web 应用程序的程序员的生活更加轻松一些。而 Struts 的出乎意料地流行说明了 Struts 试图解决的问题不仅仅是我一个人碰到过，Struts 解决了很多普遍存在的问题。

当 JSP 的最初规范出台的时候（0.91 版和 0.92 版），在该文档中一个引人入胜的概念在于两种不同的构造基于 JSP 的应用程序的基本设计风格。Model1 的特性是表单提交的结果是由产生该表单的 servlet 或者 JSP 页面来处理。这种设计风格使得我们将表现层逻辑（用来创建表单的）和业务层逻辑（用来对输入数据进行校验以及处理输入的数据）混和在一起。当项目里面的程序员仅仅具有一方面的技能（或者是稍微懂一些程序设计的页面设计人员，或者是仅仅懂一些 HTML 的 Java 开发人员）的时候，经常使用这种设计方法。同时，当项目的时间很紧张（例如，原型系统需要在下个星期一的时候运行，否则不能够得到投资基金的支持），使用该方式也很有用。但是经验告诉我，Model1 的设计方法会导致在以后的系统维护和增强时都很困难。

与此相反，Model2 的设计方法是将表单提交的数据交给一个控制器模块。控制器模块将数据转发给合适的业务逻辑模块来处理。业务逻辑模块和数据库打交道，同时还获取要进一步和用户打交道使用的数据。然后，控制器模块将创建响应页面的工作交给表现层模块来处理。表现层模块的唯一功能就是创建响应页面。

你可能认为使用 Model2 风格的程序可能要更加复杂——特别对于一个简单的应用程序而言，这种复杂性会使得项目没有办法进行。实际上，基于 Model2 创建的应用程序并不比基于 Model1 创建的应用程序需要更多的时间。但是使用 Model2 创建应用程序的好处会很快显现出来。如果我们基于一个恰当的体系结构来创建应用程序，则对于一个层次进行的主要修改并不会（或者很少）对于其他层次有影响。这样我们就可以立刻重用那些并没受到影响的层次中的逻辑。

在我对于基于 Web 应用程序的体系结构进行这样探索的时候，我的职业生涯也同时带我进

入了一个很有趣的方向。当时，我在一个为美国长途汽车行业提供信息服务的公司工作，该公司期望将他们的服务也扩展到欧洲。这就使得他们的应用程序需要具有处理多种语言和国际化的能力。我很快地编写了一个简单的控制器 servlet，使得我们可以实现简单的 MVC 体系结构，但是并不奏效，因为我们需要的是可以进行语言选择的控制器。

我在国际化开发中最开始的工作就是使用 JSP 1.1 提供的新的 JSP 标签来创建“用户界面组件”。该工作最终演变成了如今 Struts 中的 `<bean: message>` 标签。

在这之后，我加入了 Sun 公司。负责开发 Tomcat servlet 和 JSP 容器（作为 Tomcat4 中 Catalina servlet 容器的主要设计师）。该开发工作中的一大部分是开源项目 Apache 的一部分。当 Sun 在 1999 年将其对于 JSP 和 servlet 的参考实现贡献给 Apache 后，我们关于 servlet 容器的开发就是 Jakarta 项目的一部分。但是，当时我对于 Model2 中原始的应用程序设计方式并不满意，因此我决心为此做些工作。

尽管我对于如何解决该问题有着很好的想法，但是真正的代码直到 2000 年的 Memorial Day（阵亡战士纪念日）周末休假时才出来（我妻子一定对此感到很委屈，因为我带着我的笔记本和全家人一起去 Oregon 海滩度周末）。而 ActionForm 的第一个版本就是在那个假期里产生的。而该想法很明显解决了很多设计上的问题。除此之外，在表现层以及业务逻辑组件上定义逻辑名字，和通过一个配置文件来集中管理这些名字，可以避免两层开发人员之间的工作重叠以及避免对一层的修改影响其他层的问题。

我在 Tomcat 项目上的经历使我知道开源项目开发的好处，因此很自然我将 Struts 贡献给了开源社区。这个选择以及 Struts 自身具有优美地解决开发 Web 应用程序中的根本问题的能力，使得 Struts 以惊人的速度被人们接受并使用。全世界成千上万的程序员下载了 Struts，通过学习曲线，通过 Struts 的邮件列表开始问问题（也得到回答），然后成功地使用 Struts 构造应用程序。

当然，不可能通过我一个人的力量来实现这些全部功能。Ted、Cedric、David 以及 Struts 项目开发委员会中的其他新老成员，还有 George 以及 Struts 开发团队，才使得 Struts 框架结构能够比我一个人做得更加成功。对于他们，我发自内心地感谢。对于你们，本书的读者，我希望你们会发现在 Struts 技术和 API 上花时间是值得的。

Craig McClanahan

Struts 框架结构的架构师及主要开发人员

前言

到 2000 年的时候，Java 已经是一个成熟的语言了。早期版本中的问题已经被解决，同时一些很有趣的开发工具和类库也开始出现在市场上。我当时已经有过数年开发 Web 应用程序的经验。和许多其他程序员一样，最初开发 Web 应用程序的时候是使用 JavaScript 和 Perl 语言。这种组合具有强大的功能，但是维护它简直就是一场噩梦。后来又使用过 ColdFusion，它同样具有强大的功能，但是对于我的客户而言，它太贵了。我曾经还试过 FileMaker Pro，这是一个很有趣的工具，但是它过于专业化了。

我当时做的这一系列 Web 应用程序的主要客户是一个公共广播电台。该电台的主要经济支柱是每年的拍卖活动。本地的供应商提出自己的商品和服务，而人们通过广播电台来投标并支持该电台。当然，对于那些高端的拍卖品，例如手工艺品、汽车、度假产品、名人签名等等，我们同时也会在网站上面贴出相关的图片。

在 1998 年的时候，对于那些高端拍卖品，我们使用由 JavaScript 和 Perl 语言开发的一个应用程序来进行“预投标”。那些真正的投标还是通过电台的实况投标活动进行。应用程序真正要做的仅仅是得到开始投标的价格。到 1999 年的时候，我们开始同时使用在线以及电话投标两种形式了。因为发现每年原来使用的平台并不能完全符合我的要求，我都不得不使用另外的平台来发布被拍卖的产品。

对于当时正在使用的 Apache 服务器很满意，因此我花费了一些时间来研究刚刚诞生的 Jakarat 站点。在那个站点上，我发行了 Struts。最初，我甚至都觉得 Struts 项目可能已经不是一个活动的项目了。但是它的文档看起来是一个很有潜力的项目。因此，我订阅了它的邮件列表，看看是否还有人工作在该项目上。该项目的文档中附带了一个例子应用程序，因此我便一个人来试试该例子应用程序，试图搞清楚该框架结构到底用来干什么。这项研究工作最终演变成了“Struts 应用程序的介绍”这样一篇文档，其中通过一系列的截屏来说明例子应用程序是如何工作的。我将该文章发表在邮件列表上，其他一些订阅了该邮件列表的人指出了我文章中的一些小错误。

我继续订阅该邮件列表，尽我所能帮助其他人，同时也得到其他人的帮助，该邮件列表的使用人数急剧上升。到年底的时候，Struts 的体系架构师 Cragi McClanahan 希望有人可以帮助他完成 Struts 1.0 版本的文档工作。我被 Struts 小组选中来完成该文档工作。到 2001 年，我们终于成功发布了 Struts 1.0。

在这以后，我开始了“More About Struts”的页面制作工作。最初，我仅仅是用它来存放那些我编写的和 Struts 相关的材料，后来我又在其中添加了链接其他人编写的关于 Struts 的材料。渐渐地，越来越多关于 Struts 的文章出现了。我的关于 Struts 的资源列表的页面也越来越大，同时也越来越受到欢迎，因此我将它移到 Struts 的主站点上。现在，它包含了一系列的页面，链接我们知道的全部和 Struts 相关的材料上。

Struts 的邮件列表中包含了很多很有用的内容。特别在开始的时候, Craig 也经常通过它来给出实现细节以及体系结构上的指导。但是, 现在想在归档后的邮件列表中找到合适的回答已经变成了一个挑战。因此, 我又开始对邮件列表里的每一个问题进行归纳, 找到最好的部分。这项工作最后成为了一个关于 Struts 的相当大的“FAQ”。2001 年的 6 月, JGurn 决定为 Struts 也设定一个论坛和 FAQ, 因此我们将该 FAQ 移交给 JGurn, 但是我还是负责该 FAQ。

同时, 许多出版社也开始注意到 Struts。很多出版社向我约稿。在咨询了其他的 Struts 社团后, 我们最终决定和 Manning 出版社一起合作。和 Apache 一样, Manning 对于开源项目也是有着长期的质量承诺。尽管我们期望这本书尽快面世, 但是我们也期望保证本书尽可能成为最好的。

结果就是我们现在看见的本书。它其实是一本合作的结晶。David Winterfeldt, Struts Validator 的创始人, 很热心地负责了关于 Validator 的章节。同样, Cedric Dumoulin, Tiles 的创始人, 也负责 Tiles 的章节。George Franciscus 负责第 1 章, 也是最关键的一章。第 1 章专门针对那些刚刚进入该领域的新手写的。我们甚至请 Craig 为本书写了序言 (Craig 更喜欢写程序)。当然, 其他的 Struts 开发人员和整个社区, 都在不同的阶段审阅了本书的手稿。我们在此对于那些建议表示感谢。

我们就要进入使用 Struts 的第三个年头了。现在我们不是每年都重写一次, 但我们每年都将改进它。

Ted Husted
Fairport 纽约

关于本书

Struts 框架结构包含了很多相关技术，使得开发 Web 应用程序的程序员可以容易地开发一个基于标准的，具有扩展性和维护性的应用程序。无论是新手还是老手，Struts 已经成为开发框架的当仁不让的选择。

本书是一本手把手教授 Struts 框架结构的书。本书除了文字外还有很多的实例应用程序研究，这些应用程序都是专门设计用来说明如何使用 Struts 的最好经验。因此本书特别适用于那些职业程序员，可帮助读者得到实际的经过检验的建议来构造最符合 Struts 要求的应用程序。

当程序员使用 Struts 构造应用程序的时候，通常会使用一些相关的技术。要想全面介绍这些技术，可能需要好多卷才行。为了保证能够在一本书的范围内来讲清楚 Struts，我们并不试图描述 HTML 语言、JSP 的语法、JavaBean 的开发方法以及类似的技术细节。我们假设本书的读者已经对以上的知识有了充分的了解，从而可以理解我们给出的例子。同时，我们也假设读者已经了解 URL、文档超链接、Web 应用程序的压缩包，以及其他创建和发布 Web 应用程序的知识。

我们同样没有涉及 Java 程序设计语言。对于 HTML、JSP、JavaBean 以及一些其他相关技术的介绍，现在市场上有很多很不错的书。我们假设阅读本书的读者已经对于 Java 语法、开发周期以及面向对象的程序设计都有所了解，对于关系型数据库的大致了解，特别是有些 JDBC 的实际开发经验更好。

当然，本书的重点是 Web 应用程序开发以及 Struts 框架结构。前面提到那些技术（HTML、Java、数据库等）之间的相互交互才是本书的重点，也是本书要深入介绍的部分。

但是，对于 Struts 依赖的技术并不十分了解的读者而言，本书包含对于 HTTP、Java Servlet、JSP 以及定制 JSP 标签的介绍。

简单介绍

第 1 章从总体上介绍了 Web 应用程序的开发，特别介绍了 Struts。我们简单介绍了 Struts 是如何编写分发的，用来构造基于 Web 应用程序的必备技术，以及 Struts 的大致体系结构。在该章的最后，介绍使用 Struts 开发一个简单的 Struts 应用程序。

第 2 章研究了 Struts 体系结构。我们先对于 Struts 的体系结构进行了一个总体上的介绍，然后仔细研究了如何通过框架结构来控制流程。该章还直接对于 Struts 的优点和缺点进行了说明。该章主要是想给那些真正的程序员一个关于 Struts 到底是什么的基本认识，同时，也可以帮助项目管理人员决定是否在项目中使用 Struts。

第 3 章介绍了开发一个简单应用程序的过程。和第 1 章的练习类似，这是一个简单的登录程序，但是其中包括了大多数 Web 应用程序都有的特点。该章的目的是在读者进入到第二部分的细节前，通过分步练习对于整个 Struts 有个很好的了解。为了更加接近现实，我们将整个

应用程序从 Struts1.02 升级到 Struts1.1。

第4章介绍了 Struts 框架结构的骨架——配置文件，同时我们也描述了如何配置 Web 部署描述文件以及用来构造和部署应用程序的 Ant 文件。

第5章介绍 ActionForm。ActionForm 是 Struts 中的一个关键对象，它可以实现多种功能：数据传输、防火墙、API、数据校验以及类型转换。我们会介绍一些技术来使得读者可以更好地利用表单 Bean。

第6章介绍了 ActionForward 对象。Web 应用程序的最有趣的地方就在于用户可以从应用程序某个地方跳转到另外一个地方。而 ActionForward 可以用来定义应用程序的每个进入点，通过 ActionForward 可以很容易地看出应用程序是否已经覆盖了全部的需求。

第7章介绍了 ActionMapping 对象。它是 Struts 控制器的集成，Action 类可以设计成为可以复用的类，然后通过 ActionMapping 的定义来配置它，实现不同的功能。在该章我们会介绍如何使用 ActionMapping 来控制应用程序的流程以及如何更大限度地利用 Action 类。

第8章介绍 Action 类。Action 类是 Struts 中真正干活的类，同时也是程序员用得最多的类。我们会详细介绍 Struts 中捆绑的一些 Action 类以及 Scaffold 包中的一些最有用的 Action 类。同时讨论使用 ActionForm 来填充业务层对象的问题。

第9章介绍了 ActionServlet。该控制器类是整个框架结构的代言人，它负责调用其他的模块来让其他模块实现具体的功能。此外我们还介绍了如何使用新的方法来对 ActionServlet 进行定制，以满足其可以更好地使用应用程序或者某个应用程序模块的需求。

第10章介绍了 Struts 的标签库以及服务器端页面。从用户的角度来看，Web 页面就是应用程序，并且表现了应用程序的能力。使用 Struts 的一个关键好处就在于它可以将获取内容和显示内容分隔开来。在该章中，我们仔细介绍 Struts 的标签，同时还简单说明了 Struts 可以使用的其他表现层技术：XSLT 和 Velocity 等。尽管大多数基于 Struts 的应用程序都是依赖于 JSP 的，但是 Struts 框架结构本身可以和任何基于 Java 的表现层一起工作。

第11章介绍了 Tiles 框架结构。类似于 Tiles 这样的动态模板系统，使得一个基于 Web 的应用程序的表现层也具有类似于编程的模式。一个 tile 封装了一段 HTML 的语句就如同一个 Java 方法封装了一段 Java 代码一样。通过 Tiles 来构造 Web 页面，使得随意的混乱的构造 HTML 变成了一个统一而又灵活的过程。

第12章介绍了如何对于用户的输入进行校验。这是一个很重要的话题。一个很常用的对于 Struts 核心的扩展就是 Struts Validator。这是一个很强大的模块，通过一个配置文件，可以同时提供服务器端和客户端的校验。我们会说明如何使用预先定义的经验器以及自己编写的校验器，将校验器集成到应用程序中的过程。

第13章介绍 Struts 国际化的问题。Struts 从一开始就对国际化提供支持。该章介绍了在 Struts 中是如何实现国际化的，以及哪些模块和国际化相关。同时还介绍了如何编写程序才能使得自己的应用程序具有国际化特性。一个基本的原则就是我们必需可以先用一种语言来开发应用程序，但是以后可以很容易添加对于另外语言的支持。

第14章介绍如何将 Struts 和数据服务连接起来。该章介绍了如何使用辅助类来将 Struts 的 Action 类和不同类型的企业级数据系统（数据库、搜索引擎和基于内容的服务）连接起来。我

们提供了使用 JDBC、Lucenc 和 Rich Site Summary 为数据服务的例子。

第 15 章说明我们的例子程序——Artimus。该应用程序覆盖了全部的内容。它将 Struts 以及那些插件的关键特性放置到一个应用程序包中。认证、定制、本地化、Scaffold、Tiles、Transaction、Validator 等等，这些 Struts 提供的全部功能都在该例子中体现了。

第 16 章是 Struts1.1 的升级指南。我们使用第 15 章的 Artimus 应用程序的例子，并增加了 Struts1.1 的新特性：DynaForm、插件 Action、多应用程序模块。如果读者的应用程序是基于 Struts1.0，并且想进行升级，则请参见该章。

第 17 章介绍如何在 Struts 中使用 Velocity。我们重新审视一遍第 3 章的登录应用程序并使用 Velocity 模板改造它，同时还比较了 Velocity 和 JSP。

代码

本书的例子程序对应的源代码已经贡献给 Apache Software Foundation。该例子已经是 Struts 发行版中的一部分，同时也可以从 Manning 出版社站点下载（www.manning.com/husted）。

在本书前面部分中使用的大部分源代码都是代码片断，是用来为文字进行说明的，当给出完整源代码时，我们将每个例子编了号，代码中的注释也具有有一些编号，对于那些要特别注意的地方使用黑体字加以注明。

其他说明如下：

- 因为可能有多于一种形式的方法调用，因此在文字中出现的对于方法的引用不会包含签名。
- 对于 JSP 标签的引用会包含括号和默认的前缀，但是没有包括该标签可以使用的属性（`<bean:write>`）。
- 在文本中对于 XML 元素的引用会包括括号，但是不包括属性以及该元素的结束标签（`<action>`）。
- 当在某一节中第一次使用某个 Java 类或者标签库的时候，则在括号中包含完全的包名。
- 在代码片断或者例子程序中，对于混和的 JSP 和 HTML，我们对于 HTML 使用大写字母，对于 JSP 使用小写字母。

参考文献

参考文献是通过方括号来表示的。例如 [ASF, Aritmus]。本书后面的参考文献包含完整的相关出版物以及对应的 URL。

作者在线

如果购买了本书，则可以进入由 Mannish 出版社运行的一个私有 Web 论坛来发表关于此书的评论，咨询技术问题，从其他读者或者作者处获取帮助。要访问该论坛，请打开 <http://www.manning.com/husted>。该页面提供了如何在注册后进入论坛的介绍、可以提供什么样的帮助以及该论坛的规章等。

关于作者

Ted Husted：一位知名的 Struts 权威，Struts 开发小组的一个活跃分子，JGuru 中 Struts 论

坛的管理员。作为一名咨询师，Ted 曾在专业 Struts 开发小组中工作。Ted 同时也负责帮助维护 Apache jakarta 项目，该项目包含了 Struts 框架结构。Ted 和他的妻子、两个孩子、四台电脑和一个小猫一起住在纽约。

Cedric Dumoulin：是 Struts 开发小组的一个活跃分子，同时也是 Tiles 框架结构的设计师。Cedric 是 Lille 大学的一名研究人员，同时也在一家领先的网络银行公司的研发部门工作。他住在法国的 Lille。

George Franciscus 是 Nexcel 的领导。Nexcel 对于多个行业提供技术和管理的咨询服务，这些行业包括电信、银行、人寿保险、财产保险和意外保险。George 在 Java、J2EE、Domino、关系型数据库以及大型机领域都有很多经验。他从多伦多大学获得了计算机科学专业的学士学位。George 和他的妻子和三个孩子一起生活在多伦多的 Ontario。

David Winterfeldt 是 Struts 委员会的一员，同时也是 Commons Validator 包的作者。他是一家实现 J2EE 技术的大公司里的高级程序员。David 现在住在纽约市。

Craig McClanahan：Struts 框架结构的创建者，本书的序言就是他写的。Craig 是 Tomcat 4 的设计师同时也是 Java Web Service 开发包的实现设计师。他现在是 Sun 关于 JSF 的规范设计以及基于 J2EE 平台的 Web 层体系结构的领导人。Craig，作为 Struts 的主要开发者，提供了本书最重要的部分——Struts 框架结构。

关于封面图片

本书封面图片是一个来自 Bordeaux 荒原上的牧羊人（Berger des Landes de Bordeaux）Bordeaux 地区在法国的西南部，是一个日照充分的丘陵地区，既是一个理想的葡萄园地区，也有很多沼泽以及点缀其中的小农场和一群群的羊。牧羊人踩在高跷上，使得自己可以很容易地在沼泽中行走，并且照看自己的羊群。

该图片取自一本介绍法国旅行的图书 *Encyclopedie des Voyages*，由 J. G. St. Saver 在 1796 年出版。在那个时候，出去旅行放松才刚刚出现。因此，如同 *Encyclopedie des Voyages* 这样的小册子很流行。既满足出门旅行者的需要，也可以使得那些不在该地区的法国人或者外国人可以在家里对其他地方有所了解。

Encyclopedie des Voyages 中有很多插图，生动地说明了 200 年前世界上城镇和省之间的独特性和个性，两个就算是相隔几十英里的人也可以很容易地从他们的着装方式上加以区分。这些旅行手册使得我们了解到曾经有段时间（甚至是不久之前），人们的生活是多么的隔绝，现在我们的生活则是如此紧密相连。

从那以后，着装方式慢慢改变，那些不同区域间的显明差异也慢慢消失。现在已经很难分辨一个人是来自哪个大陆了。从乐观角度看，现在的人在文化和着装上有自己的特点，或者说现在的生活更加多元化，充满乐趣和智慧。

因此，本书封面通过一个 200 年前旅行手册上的图片来说明当时区域间的差异，生动反衬了当前创造性、主动性的工作，以及计算机给我们带来的生活乐趣。

目 录

序言

前言

关于本书

第一部分 Struts 起步

第 1 章 概述	1
1.1 本书的目的	1
1.1.1 Struts 的开发人员	1
1.1.2 为什么说 Struts 是开源软件	2
1.1.3 为什么被称为 Struts	2
1.2 应用程序框架的概念	2
1.3 必备知识	3
1.3.1 HTTP	3
1.3.2 通用网关接口	4
1.3.3 Java servlet	5
1.3.4 JSP	6
1.3.5 JSP 标签	7
1.3.6 JavaBean	8
1.3.7 Model 2	9
1.4 Struts 概述	9
1.4.1 构造一个简单的应用程序	11
1.4.2 准备工作	11
1.4.3 如何完成整个练习	12
1.4.4 回顾	17
1.5 总结	20
第 2 章 研究 Struts 体系结构	21
2.1 言众人所言	21
2.2 为什么使用 Struts	21
2.2.1 历史的回顾	21
2.2.2 Struts 介绍	22
2.2.3 Struts 控制层	22
2.2.4 使用 Struts 开发 Web 应用程序	26
2.3 为什么需要框架结构	26

2.3.1 Web——麻烦的源头	26
2.3.2 servlet 解决方案	27
2.3.3 Servlet 框架结构	28
2.3.4 黑盒和白盒系列	29
2.4 Struts、Model 2 和 MVC	29
2.4.1 MVC 的演变	29
2.4.2 Model 2 的产生	30
2.4.3 应用程序层——独立的视图层	31
2.4.4 Struts 如何实现 Model 2、MVC 和分层	32
2.5 Struts 控制流	33
2.5.1 流程概述	33
2.5.2 更微小的细节	35
2.5.3 Struts 的性能	37
2.6 Struts 的强项和弱点	38
2.6.1 Struts 的缺点	38
2.6.2 Struts 的优点	40
2.7 总结	41
第 3 章 构造一个简单的应用程序	42
3.1 从基础了解 Struts	42
3.2 简述登录应用程序的流程	43
3.2.1 起步	43
3.2.2 将会使用的页面	43
3.2.3 欢迎页面	44
3.2.4 登录页面	44
3.2.5 再次进入欢迎页面	45
3.2.6 退出欢迎页面	45
3.2.7 特性摘要	45
3.3 解剖登录应用程序	46
3.3.1 浏览器中欢迎页面的代码	46
3.3.2 欢迎页面的 JSP 源代码	47
3.3.3 欢迎页面的配置信息	49
3.3.4 浏览器中登录页面的代码	50
3.3.5 登录页面对应的配置部分	52
3.3.6 LogonSubmit 的源代码	52

3.3.7 LogonForm 的源代码	53	4.4.9 编写自己的子类	92
3.3.8 LogonAction 的源代码	55	4.4.10 Struts 配置文件的一个框架	92
3.3.9 LogoffAction 的源代码	60	4.5 应用程序的资源文件	94
3.4 构造一个应用程序	62	4.6 Ant 文件	95
3.4.1 定义需求	62	4.7 配置 Struts 的核心	97
3.4.2 计划应用程序	63	4.7.1 安装 Java 和 Servlet 容器	97
3.4.3 计划源代码树结构	65	4.7.2 安装一个开发环境	98
3.4.4 安装自己的开发工具	65	4.7.3 安装 Struts 的核心文件	98
3.4.5 建立 build.xml 文件	66	4.8 配置 Tiles 框架结构	98
3.4.6 建立 web.xml 文件	66	4.9 配置 Struts Validator	99
3.4.7 建立 struts-config.xml 文件	67	4.10 使用 Struts 空白应用程序	100
3.4.8 测试部署	68	4.11 配置模块化的应用程序	102
3.4.9 构造我们的欢迎页面	69	4.11.1 分而治之	102
3.4.10 构造登录页面	70	4.11.2 前缀化页面	104
3.4.11 构造常量类	71	4.11.3 更新配置文件	104
3.4.12 构造其他的类	72	4.12 共享 Struts 的 JAR 文件	104
3.4.13 创建用户目录	72	4.13 总结	105
3.4.14 配置 ActionErrors	73		
3.4.15 编译和测试登录页面	73	第二部分 提出自己的框架结构	
3.4.16 修改欢迎页面	74		
3.4.17 Struts 的 Action: Forward- Action	75	第 5 章 处理 ActionForm	107
3.5 总结	76	5.1 输入垃圾, 输出珠宝	107
第 4 章 配置 Struts 组件	77	5.2 多面手 ActionForm	110
4.1 三个 XML 文件和一个属性文件	77	5.2.1 使用 ActionForm 来填充 自己的域	110
4.2 Web 应用程序部署描述文件	78	5.2.2 使用 ActionForm 作为 数据缓存	111
4.2.1 web.xml 文件	78	5.2.3 使用 ActionForm 作为数据 校验器	112
4.2.2 ActionServlet 的参数	80	5.2.4 使用 ActionForm 作为类型 转换器	112
4.3 Struts 配置	81	5.2.5 使用 ActionForm 作为数据 传递对象	112
4.3.1 细节	82	5.2.6 使用 ActionForm 作为防火墙	113
4.3.2 管理修改	83	5.3 ActionForm 的设计结果	114
4.3.3 原则: 隐藏变化	83	5.3.1 ActionForm 可能和业务逻辑层 对象共享名字	114
4.4 Struts 配置元素	84	5.3.2 ActionForm 可能减少定制的 代码量	114
4.4.1 <global-exceptions>	85	5.3.3 ActionForm 可以封装 辅助函数	115
4.4.2 <form-beans>	86	5.3.4 ActionForm 可以内含	
4.4.3 <global-forwards>	87		
4.4.4 <action-mappings>	88		
4.4.5 <controller>	89		
4.4.6 <message-resources>	90		
4.4.7 <plug-in>	90		
4.4.8 <data-sources>	91		

其他 bean	115	7.1.2 ActionMapping 目录	142
5.4 ActionForm 的其他类型	116	7.2 ActionMapping 属性	143
5.4.1 Map-backed ActionForm	116	7.2.1 path 属性	143
5.4.2 DynaActionForm	117	7.2.2 forward 属性	144
5.5 哪些情况下不使用 ActionForm	118	7.2.3 include 属性	144
5.5.1 为什么 ActionForm 不是一个 Map 对象	118	7.2.4 type 属性	144
5.5.2 为什么 ActionForm 不是一个 简单的 JavaBean	118	7.2.5 className 属性	144
5.5.3 为什么 ActionForm 不是 一个接口	119	7.2.6 name 属性	145
5.6 使用 ActionForm	119	7.2.7 roles 属性	145
5.6.1 实现业务层逻辑	120	7.2.8 scope 属性	145
5.6.2 内含可改变的数据对象	121	7.2.9 validate 属性	145
5.6.3 设置不可改变数据对象	122	7.2.10 input 属性	145
5.6.4 设置可改变的数据对象	123	7.2.11 parameter 属性	146
5.6.5 使用工厂方法	123	7.2.12 attribute 属性	147
5.6.6 传递一个 Map 对象	124	7.2.13 prefix 和 suffix 属性	147
5.6.7 通过反射传递数据	126	7.2.14 unknown 属性	147
5.6.8 使用接口类	130	7.3 内部组件	148
5.7 BaseForm	131	7.3.1 本地转发	148
5.7.1 SessionLocale	131	7.3.2 本地异常	148
5.7.2 Dispatch	132	7.4 编写自己的 ActionMapping	148
5.7.3 自动填充	132	7.5 总结	149
5.7.4 BaseMapForm	132	第8章 使用 Action 对象	150
5.8 总结	133	8.1 准备, 设定, 行动	150
第6章 使用 ActionForward 对象	134	8.2 使用 Action 对象	150
6.1 ActionForward 的功能	134	8.2.1 Action 是什么	151
6.2 ActionForward 如何工作	135	8.2.2 何时调用 Action	152
6.3 本地转发和全局转发	136	8.2.3 Action 可以做什么	153
6.4 运行时刻参数	137	8.2.4 Action 到底是什么	157
6.4.1 在页面中增加参数	137	8.3 标准的 Action	158
6.4.2 在 Action 类中添加参数	138	8.3.1 标准的桥接 Action	159
6.5 动态转发	138	8.3.2 标准的基本 Action	161
6.6 导航条内容不变的原因	138	8.4 级联 Action	165
6.7 编写自己的 ActionForward	139	8.5 Scaffold Action	167
6.8 总结	140	8.5.1 仅仅和转发相关的 Action	167
第7章 ActionMapping 的设计	141	8.5.2 辅助 Action	172
7.1 了解 ActionMapping	141	8.6 基本的 View Action	174
7.1.1 ActionMapping bean	142	8.7 辅助 Action 类使用的技术	175
		8.7.1 可选择的转发	176
		8.7.2 先调用	177
		8.7.3 捕获级联的异常	177
		8.7.4 灵活的错误信息转发	179

8.7.5 提示成功消息	180	第 11 章 使用 Tiles 开发应用	235
8.7.6 可变换的视图	180	11.1 管理布局	235
8.7.7 使用反射技术来调用方法	181	11.1.1 用动态模板进行分层	235
8.7.8 使用反射来调用类	181	11.1.2 运用模板的结果	236
8.8 使用灵活转发	182	11.1.3 使用模板	236
8.9 总结	186	11.1.4 集成模板、Tiles 和 Struts	237
第 9 章 扩展 ActionServlet	187	11.2 创建一个 layout 模板	238
9.1 ActionServlet 的地位	187	11.2.1 tile 是什么	240
9.2 RequestProcessor	189	11.2.2 部署 Tiles 模板	241
9.2.1 process 方法	190	11.2.3 添加样式表	242
9.2.2 processRoles	190	11.2.4 模板和 MVC	243
9.3 ExceptionHandler	192	11.3 Tiles Definition	243
9.4 插件	193	11.3.1 声明 Definition	243
9.5 总结	193	11.3.2 JSP 声明	244
		11.3.3 配置文件声明	247
		11.3.4 把 Definition 作为 ActionForward	249
第三部分 构造自己的页面		11.4 Tile 属性	250
第 10 章 显示动态内容	195	11.4.1 useAttribute	250
10.1 标签——就是你	195	11.4.2 importAttribute	251
10.1.1 JSP 标签库——有什么优点	195	11.4.3 put	251
10.1.2 Struts 和 JSTL	198	11.4.4 putList 和 add	253
10.1.3 Struts 标签库和 MVC	199	11.5 把一个应用程序转化为 Tiles	253
10.2 使用扩展标签库	200	11.5.1 设置 Tiles 框架	253
10.2.1 扩展标签是如何工作的	200	11.5.2 测试默认配置	254
10.2.2 如何安装扩展标签	202	11.5.3 浏览页面	254
10.2.3 扩展标签库不是什么	203	11.5.4 使用 <tiles: insert> 重构页面	256
10.3 Struts 标签库	204	11.5.5 把 <tiles: insert> 标签析取 到 Definition 中	262
10.3.1 Struts 标签库的常用功能	205	11.5.6 模式化基础 layout	265
10.3.2 bean 标签	206	11.5.7 把 Definition 精练为基础类 和扩展类	266
10.3.3 html 标签	208	11.5.8 形成例程	267
10.3.4 logic 标签	210	11.5.9 对移植的管理	267
10.4 使用 Struts 标签	213	11.6 总结	268
10.4.1 Struts 的标签	213	第 12 章 验证用户输入	270
10.4.2 基础	213	12.1 只有看见时我才知道	270
10.4.3 技术	220	12.1.1 我们不能拒绝的输入	270
10.4.4 可用的控件	231	12.1.2 Web 层的校验	271
10.5 不同的视图	232		
10.5.1 Struts 和 JSP	232		
10.5.2 servlet 上下文	232		
10.5.3 除了 JSP 以外	233		
10.6 总结	233		

12.1.3 使用校验器的结果	272	12.11.7 在 ApplicationResources 中添加 新的选项	300
12.2 Struts Validator 简述	273	12.11.8 调用 Struts 校验器	301
12.3 基本校验器	280	12.11.9 测试和重复	301
12.3.1 required 校验器	281	12.11.10 删除 ActionForm 子类	302
12.3.2 mask 校验器	281	12.12 总结	303
12.3.3 range 校验器	282	第 13 章 内容本地化	305
12.3.4 maxLength 校验器	283	13.1 用另外一个名字	305
12.3.5 minLength 校验器	283	13.1.1 为什么进行本地化	306
12.3.6 byte、short、integer、long、 float 和 double 校验器	283	13.1.2 Java 的国际化功能	306
12.3.7 date 校验器	283	13.2 Struts 的国际化组件	310
12.3.8 creditCard 校验器	284	13.2.1 Session 中的 Locale 属性	310
12.3.9 email 校验器	284	13.2.2 MessageResources	311
12.4 资源包	284	13.2.3 默认的资源包	311
12.4.1 默认的资源包	285	13.2.4 ActionErrors	313
12.4.2 默认校验器消息	285	13.2.5 ActionMessages	313
12.4.3 定制化的校验器消息	286	13.2.6 区域敏感的 JSP 标签	314
12.5 配置文件	286	13.3 本地化 Struts 应用程序	318
12.6 校验器的 JSP 标签	287	13.3.1 允许本地化	318
12.7 ValidatorForm 和 Validator ActionForm	290	13.3.2 使用框架结构 Locale 对象	319
12.8 本地化的校验器	291	13.3.3 将标签和消息存放在属性 文件中	320
12.9 可插入的校验器	291	13.3.4 创建语言相关的属性文件	320
12.10 技术	292	13.3.5 在可以本地化模块中指定恰当 的键值	321
12.10.1 多页面校验	293	13.3.6 其他模块使用 <bean:message>	321
12.10.2 取消按钮	293	13.4 本地化其他模块	321
12.10.3 定制化的消息	293	13.4.1 本地化 Struts Validator	321
12.10.4 相关联域	294	13.4.2 本地化 Tiles	322
12.10.5 联合使用校验器和 validate 方法	295	13.4.3 本地化集合	322
12.11 移植应用程序到 Struts Validator	296	13.5 总结	323
12.11.1 安装配置 Validator 框架结构	296	第 14 章 使用数据服务功能	325
12.11.2 测试默认的配置	297	14.1 起步	325
12.11.3 回顾你的校验方法	297	14.1.1 从层次的角度看 JDBC	325
12.11.4 扩展 ValidatorForm 或者 Scaffold BaseForm	298	14.1.2 介绍数据服务	326
12.11.5 选择一个校验器来进行 移植工作	298	14.2 研究业务层	327
12.11.6 添加 formset、form 和 field 元素	299	14.2.1 Struts——使用自己的模式	327
		14.2.2 定义业务对象	327
		14.2.3 设计业务对象	328
		14.2.4 设计结果	329

14.2.5 混合业务方法和 Action (不推荐)	329	15.4.7 受保护的 URL	359
14.2.6 一个简单的例子	330	15.4.8 授权用户	359
14.3 在 Struts 中使用 ProcessBeans 和 JDBC	330	15.4.9 鉴证策略	359
14.3.1 介绍 ProcessBean	331	15.5 ArtimusServlet	359
14.3.2 作为数据传递对象的 ProcessBean	332	15.5.1 子类	361
14.3.3 填充 ProcessBean	333	15.5.2 串标志符	361
14.3.4 执行 ProcessBean	333	15.5.3 扩展点	361
14.3.5 访问数据服务	334	15.6 应用程序与 SQL 属性文件	362
14.3.6 使用典型的流程	335	15.7 index.jsp	363
14.3.7 对业务行为进行编码	336	15.8 全局转发	363
14.3.8 使用 ProcessBean 作为 持久层	338	15.9 /find/Recent	365
14.3.9 使用其他的持久层	338	15.9.1 扩展 bean	368
14.4 使用 result 对象	339	15.9.2 super. execute	368
14.5 使用辅助 Action	340	15.9.3 getArticles	368
14.6 使用 Lucene	341	15.9.4 Access. findByLast 与 ResultList	368
14.7 使用内容摘要	345	15.9.5 ProcessResult	369
14.7.1 Digester 和 RSS	346	15.9.6 ProcessAction	370
14.7.2 获取和生成数据	346	15.10 tiles.xml 与 Article.jsp	371
14.7.3 RSS 摘要	347	15.10.1 useAttribute	373
14.8 在 Struts 中使用 EJB	349	15.10.2 baseStyle	373
14.8.1 Session Facade	350	15.10.3 标题	373
14.8.2 数据传递对象	350	15.10.4 Tiles	373
14.8.3 实现模式	350	15.11 result.jsp	375
14.9 总结	351	15.11.1 图例	376
		15.11.2 isResult	376
		15.11.3 RESULT	377
		15.12 article actions	380
		15.13 view.jsp	382
		15.13.1 headline	383
		15.13.2 content	383
		15.13.3 contributor	384
		15.14 form.jsp	385
		15.14.1 article content	386
		15.14.2 contributed/contributor	387
		15.14.3 article ID	387
		15.14.4 validation	387
		15.15 /do/Menu	389
		15.15.1 logon	390
		15.15.2 菜单	391
		15.15.3 我们的控件	392

第四部分 以实例介绍 Struts

第 15 章 Artimus: 实例研究	353
15.1 框架的框架	353
15.2 Scaffold——工具的产生	354
15.3 关于 Artimus	354
15.4 部署描述文件 (web.xml)	356
15.4.1 配置 Artimus	357
15.4.2 应用程序的属性	357
15.4.3 连接适配器	358
15.4.4 启动属性	358
15.4.5 其他配置设定	358
15.4.6 安全设定	358

15.5.1 useAttribute	373
15.5.2 baseStyle	373
15.5.3 标题	373
15.5.4 Tiles	373
15.6 result.jsp	375
15.6.1 图例	376
15.6.2 isResult	376
15.6.3 RESULT	377
15.7 article actions	380
15.8 view.jsp	382
15.8.1 headline	383
15.8.2 content	383
15.8.3 contributor	384
15.9 form.jsp	385
15.9.1 article content	386
15.9.2 contributed/contributor	387
15.9.3 article ID	387
15.9.4 validation	387
15.10 /do/Menu	389
15.10.1 logon	390
15.10.2 菜单	391
15.10.3 我们的控件	392