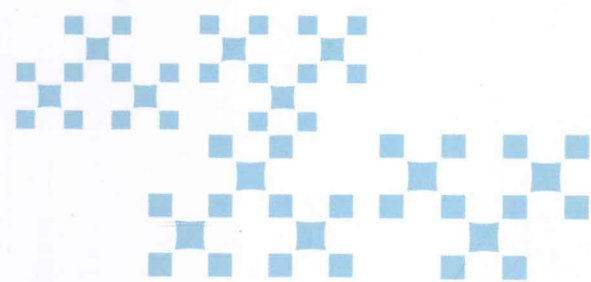


大学软件学院软件开发系列教材

Struts 2

程序开发实用教程

孙连伟 武 迪 编著



- ◆ 注重基础知识及案例的讲解。
- ◆ 教材示例源文件丰富，实例具有普遍性和实用性。
- ◆ 每章都有实训测试题和课后习题，便于巩固学习成果。
- ◆ 教材有配套的电子教案与习题答案，方便老师教学时使用。

免费赠送
PPT电子课件
及习题答案



清华大学出版社

大学软件学院软件开发系列教材

Struts 2 程序开发实用教程

孙连伟 武 迪 编著

清华大学出版社

北 京

内 容 简 介

本书从基础开始,对 Struts 2 程序设计相关知识进行了深入细致的讲解。全书共分 15 章,主要内容包括 Struts 2 技术介绍和原理、Action 的应用、核心技术的应用、Struts 2 配置文件的配置、类型转换器、拦截器、OGNL 表达式、Struts 2 的标签、Struts 2 的校验、Struts 2 的国际化、Struts 2 文件的上传下载等,最后通过留言板项目案例对前面的知识进行了综合运用。

本书不仅可以作为普通高等院校的教材,同时也是广大 Struts 2 爱好者自学的首选用书。

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 程序开发实用教程/孙连伟,武迪编著. —北京:清华大学出版社,2014
大学软件学院软件开发系列教材

ISBN 978-7-302-34004-1

I. ①S… II. ①孙… ②武… III. ①软件工具—程序设计—高等学校—教材 IV. ①TP311.56

中国版本图书馆 CIP 数据核字(2013)第 228774 号



责任编辑:杨作梅

装帧设计:杨玉兰

责任校对:宋延清

责任印制:刘海龙

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

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

课 件 下 载: <http://www.tup.com.cn>, 010-62791865

印 刷 者:清华大学印刷厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:26.5 字 数:646 千字

版 次:2014 年 1 月第 1 版 印 次:2014 年 1 月第 1 次印刷

印 数:1~3000

定 价:48.00 元

产品编号:045187-01

前 言

Struts 2 是 Struts 的新一代产品，是在 Struts 1 和 WebWork 的技术基础上进行了合并的全新的 Struts 2 框架。其全新的 Struts 2 的体系结构与 Struts 1 的体系结构差别巨大。Struts 2 以 WebWork 为核心，采用拦截器的机制来处理用户的请求，这样的设计也使得业务逻辑控制器能够与 Servlet API 完全脱离开，所以 Struts 2 可以理解为 WebWork 的更新产品。Struts 2 目前已经成为 Java EE 企业开发最实用、使用最广泛的 MVC 框架，Struts 2 框架不仅成熟、稳定，而且具有很好的扩展性，能与大部分 Java EE 框架整合。所以，Struts 2 是从事 Java EE 开发必备的技术。

1. 本书内容

本书面向 Struts 2 的实际应用开发，循序渐进地为读者介绍有关 Struts 2 开发所涉及各类知识，内容由浅入深，涵盖 Struts 2 的主要知识点，并且在介绍过程中，针对每个知识点都有相应的实例。本书通俗易懂，结构安排合理，各章主要内容如下。

第 1 章 主要介绍 MVC 的设计思想以及对 Struts 1 和 Struts 2 的概述和对比，并且介绍 Struts 2 的下载和安装。

第 2 章 讲解 Struts 2 框架开发的基本知识和步骤，例如 Struts 2 框架搭建、Action 接口访问、ActionSupport、ActionContext 类的使用。

第 3 章 主要介绍有关 Struts 2 执行的原理和流程，以及 Action 的驱动模式。

第 4 章 主要学习有关 Struts 2 中 web.xml、struts.xml 文件的配置以及 Result Types 的使用。

第 5 章 主要介绍有关 Struts 2 其他配置的使用，例如 Struts 2 中 Action 相关的标注、验证标注的使用，并介绍 struts.properties 属性文件和 struts-plugin.xml 文件的配置。

第 6 章 Struts 2 类型转换器的基本概述和 Struts 2 中内置转换器的使用，并且学习如何实现自定义类型转换器。

第 7 章 学习 Struts 2 中核心技术拦截器的原理，并了解有关拦截器的配置和使用。

第 8 章 学习表达式语言 OGNL 的语法基础和使用，例如表达式的定义、常量、如何访问 JavaBean 中的属性和方法。

第 9 章 Struts 2 中标签的使用，主要介绍 Struts 2 中通用标签、UI 标签的应用。

第 10 章 也是对 Struts 2 中标签的介绍，主要包括对 Struts 2 中 Ajax 标签和主题以及模板的介绍。

第 11 章 学习 Struts 2 中校验器的使用，主要包括对 Struts 2 中基础校验、服务器端校验、客户端校验以及 Ajax 校验的讲解。

第 12 章 学习 Struts 2 框架国际化的应用，主要包括页面内容的国际化、错误信息的国际化、日期和数值的国际化。

第 13 章 主要是 Struts 2 中综合使用技术的介绍，包括防止表单重复提交、文件上传

下载以及部分扩展技术等。

第 14 章 学习 Struts 2 和其他 Java EE 技术整合应用, 包括 Struts 2 + Spring 的整合、Struts 2 + Hibernate 的整合以及 S2SH 技术的整合应用。

第 15 章 通过一个留言板项目案例的开发, 向读者介绍留言板案例主要模块的开发, 有助于读者快速掌握 Struts 2 开发技术。

2. 本书特色

- (1) 本书在每章后面, 增加了上机实训, 以便课后加强读者的动手能力。
- (2) 每章后面提供了习题, 帮助读者温习所学的知识。
- (3) 对于后面的案例, 本书提供了完整的源文件。
- (4) 对于一些细节之处, 本书在全书需注意的地方, 增加了“注意”内容, 以便读者更好地掌握细节。

3. 适用读者

本书专门为在校学生和零基础的读者量身定制, 是普通高等院校 Struts 2 程序设计课程的首选教材, 同时也可作为 Struts 2 爱好者的自学用书。

4. 本书作者

本书由天马程序员(TOMO)俱乐部创始人之一孙连伟、天津师范大学副研究员武迪编著, 邱加永、熊萧、王学锋、张静等参与编写。编者力求表述规范、措辞严谨、通俗易懂。但限于自身水平, 疏漏之处在所难免, 读者在阅读的过程中遇到什么问题或者有好的建议或意见, 欢迎随时与我们联系。为便于相关问题的讨论, 读者可以加入 QQ 群: 259798954, 或发送邮件到 24308768@qq.com 与作者交流。

编 者

目 录

第 1 章 Struts 2 概述.....1	3.2 Action 的驱动模式 45
1.1 MVC 思想概述.....2	3.2.1 属性驱动模式 46
1.1.1 传统的 Model 1 和 Model 2.....2	3.2.2 模型驱动模式 48
1.1.2 MVC 思想及特点.....3	3.3 上机实训 49
1.1.3 常见 MVC 技术.....3	3.4 本章习题 50
1.2 Struts 1 概述 3	第 4 章 Struts 2 的配置方式(1)..... 51
1.3 Struts 2 概述 5	4.1 web.xml 文件的配置 52
1.3.1 Struts 2 的优势 6	4.2 struts.xml 文件的配置 54
1.3.2 Struts 2 的体系结构 8	4.2.1 bean 的配置 54
1.3.3 Struts 2 的配置文件 9	4.2.2 package 的配置 56
1.3.4 Struts 2 的标签库 10	4.2.3 namespace 的配置 57
1.3.5 Struts 2 的控制器 11	4.2.4 constant 的配置 58
1.4 Struts 2 的下载和安装 12	4.2.5 interceptor 的配置 58
1.5 上机实训 14	4.2.6 include 的配置 61
1.6 本章习题 14	4.2.7 action 的配置 61
第 2 章 Struts 2 入门.....17	4.2.8 result 的配置 65
2.1 HelloWorld 应用介绍.....18	4.2.9 exception 的配置 67
2.2 创建 Struts 2 的 Web 应用 18	4.3 Result Types 的配置 69
2.2.1 创建 Web 应用 18	4.3.1 Chain Result 的配置 70
2.2.2 用户请求页面 19	4.3.2 Dispatcher Result 的配置 71
2.2.3 控制器实现 21	4.3.3 FreeMarker Result 的配置 72
2.2.4 配置 Action 22	4.3.4 HttpHeaders Result 的配置 72
2.2.5 完成应用 23	4.3.5 Redirect Result 的配置 73
2.3 Struts 2 的 Action.....25	4.3.6 Stream Result 的配置 73
2.3.1 实现 Action 接口 25	4.3.7 Velocity Result 的配置 75
2.3.2 Action 访问 ActionContext 28	4.3.8 Xslt Result 的配置 75
2.3.3 继承 ActionSupport 类 30	4.3.9 PlainText Result 的配置 77
2.4 上机实训 35	4.3.10 JSON Result 的配置 77
2.5 本章习题 35	4.3.11 全局结果 81
第 3 章 Struts 2 的核心技术 37	4.3.12 动态结果映射 82
3.1 Struts 2 的执行流程和原理 38	4.4 上机实训 83
	4.5 本章习题 83

第 5 章 Struts 2 的配置方式(2)	85
5.1 Annotations 的配置	86
5.1.1 Namespace 的配置	88
5.1.2 ParentPackage 的配置	88
5.1.3 Action 的配置	90
5.1.4 Actions 的配置	91
5.1.5 InterceptorRefs 的配置	92
5.1.6 Result 的配置	93
5.1.7 Results 的配置	93
5.1.8 ResultPath 的配置	94
5.1.9 ExceptionMapping 的配置	94
5.1.10 ExceptionMappings 的配置	95
5.2 Validation Annotations 的配置	95
5.2.1 ConversionErrorFieldValidator 的配置	96
5.2.2 DateRangeFieldValidator 的配置	98
5.2.3 DoubleRangeFieldValidator 的配置	101
5.2.4 EmailValidator 的配置	102
5.2.5 ExpressionValidator 的配置	103
5.2.6 IntRangeFieldValidator 的配置	104
5.2.7 RegexFieldValidator 的配置	105
5.2.8 RequiredFieldValidator 的配置	106
5.2.9 RequiredStringValidator 的配置	107
5.2.10 StringLengthFieldValidator 的配置	108
5.2.11 UriValidator 的配置	109
5.2.12 Validation 的配置	110
5.2.13 Validations 的配置	110
5.2.14 VisitorFieldValidator 的配置	112
5.2.15 CustomValidator 的配置	114
5.3 struts.properties 的配置	115
5.4 struts-plugin.xml 的配置	116
5.5 各种配置文件的加载顺序	116
5.6 上机实训	116

5.7 本章习题	117
第 6 章 Struts 2 的类型转换	119
6.1 Struts 2 类型转换器	120
6.1.1 Struts 2 内置的类型转换器	120
6.1.2 自定义转换器	120
6.2 批量类型转换	126
6.3 上机实训	129
6.4 本章习题	130
第 7 章 Struts 2 的拦截器	131
7.1 理解拦截器	132
7.1.1 理解 DRY 规则	132
7.1.2 拦截器的原理	132
7.1.3 Struts 2 内建的拦截器	134
7.1.4 struts-default.xml 里配置的拦截器	136
7.2 拦截器的配置	140
7.2.1 配置拦截器	140
7.2.2 拦截器的使用	141
7.2.3 配置默认的拦截器	142
7.3 自定义拦截器	144
7.3.1 实现拦截器类	144
7.3.2 使用拦截器	145
7.3.3 拦截方法的拦截器	146
7.4 拦截器示例	148
7.5 用 Annotation 配置拦截器	155
7.6 上机实训	156
7.7 本章习题	157
第 8 章 Struts 2 OGNL 的应用	159
8.1 OGNL 概述	160
8.2 OGNL 语法基础	160
8.2.1 OGNL 表达式	160
8.2.2 常量	161
8.2.3 操作符	161
8.2.4 访问 JavaBean 中的属性	161
8.2.5 索引访问	162
8.3 OGNL 的使用	165
8.4 Struts 2 中的 OGNL	169



8.5 上机实训	170	12.2 页面内容的国际化	292
8.6 本章习题	171	12.3 错误信息国际化	298
第 9 章 Struts 2 的标签(1)	173	12.4 格式化输出日期和数值	300
9.1 Struts 2 标签的引入	174	12.5 资源文件的加载方式和流程	303
9.2 通用标签	176	12.6 上机实训	306
9.2.1 流程控制标签	176	12.7 本章习题	306
9.2.2 数据标签	190	第 13 章 Struts 2 上传下载和其他技术	307
9.3 UI 标签	205	13.1 Token 防止表单重复提交	308
9.3.1 表单标签	205	13.1.1 TokenInterceptor 的使用	308
9.3.2 非表单标签	225	13.1.2 TokenSessionStoreInterceptor 的使用	313
9.4 上机实训	229	13.2 Struts 2 文件的上传和下载	316
9.5 本章习题	229	13.2.1 Struts 2 文件的上传	316
第 10 章 Struts 2 的标签(2)	231	13.2.2 Struts 2 文件的下载	325
10.1 Ajax 标签	232	13.2.3 中文文件的下载	327
10.1.1 a 标签	232	13.3 Struts 2 中文乱码处理总结	332
10.1.2 autocompleter 标签	235	13.4 页面跳转技巧	334
10.1.3 bind 标签	238	13.5 使用 SiteMesh 布局	334
10.1.4 datetimepicker 标签	241	13.5.1 SiteMesh 简介	334
10.1.5 div 标签	243	13.5.2 SiteMesh 运行原理	335
10.1.6 head 标签	246	13.5.3 SiteMesh 实例	336
10.1.7 submit 标签	248	13.6 在 Struts 2 中使用 FreeMarker	340
10.1.8 tabbedPanel 标签	251	13.6.1 FreeMarker 简介	340
10.1.9 textarea 标签	254	13.6.2 FreeMarker 快速上手	341
10.1.10 tree/treenode 标签	255	13.6.3 在 Struts 2 中使用 FreeMarker	342
10.2 Struts 2 主题和模板	259	13.7 上机实训	346
10.3 上机实训	261	13.8 本章习题	347
10.4 本章习题	262	第 14 章 S2SH 技术整合	349
第 11 章 Struts 2 的校验	263	14.1 为什么要整合 S2SH	350
11.1 校验基础	264	14.2 Struts 2 与 Spring 整合	351
11.2 服务器端的校验配置	270	14.3 Struts 2 与 Hibernate 整合	356
11.3 客户端校验配置	280	14.4 Struts 2 + Spring + Hibernate 整合	361
11.4 Ajax 校验配置	284	14.5 上机实训	366
11.5 上机实训	286	14.6 本章习题	366
11.6 本章习题	287		
第 12 章 Struts 2 的国际化	289		
12.1 常见国际化示例	290		

第 15 章 Struts 2 留言板项目案例	367
15.1 系统概述	368
15.2 系统需求	368
15.2.1 前台留言板块	368
15.2.2 管理留言板块	369
15.3 系统功能描述	369
15.3.1 浏览留言	369
15.3.2 管理员后台操作	372
15.4 系统设计	374
15.4.1 系统架构设计	374
15.4.2 业务实体设计	374
15.4.3 业务逻辑设计	375
15.5 数据库设计	375
15.5.1 E-R 图设计	375
15.5.2 物理建模	376
15.5.3 设计表格	376
15.5.4 数据表脚本	377
15.6 JDBC 封装	378
15.6.1 公共接口封装	378
15.6.2 数据库连接类	386
15.6.3 Dao 工厂类	388
15.7 功能模块实现	389
15.7.1 控制器 Action 的实现	389
15.7.2 业务逻辑层的实现	396
15.7.3 struts.xml 文件的配置	397
15.8 运行工程	397
15.8.1 使用工具	397
15.8.2 工程部署	397
15.8.3 运行程序	398
15.9 上机实训	400
15.10 本章习题	400
附录 1 部分属性设置说明	401
附录 2 Struts 2 程序开发实用教程	
习题答案	405

第 1 章



Struts 2 概述

学习目的与要求:

Struts 2 是一款优秀的基于 MVC 设计模式的应用框架,本章着重讲解 MVC 的设计思想以及 Struts 2 应用开发框架的搭建。对于初学 Struts 2 的读者来说,必须认真学习本章,了解 Struts 2,学会搭建 Struts 2 开发框架,为以后的学习打下牢固的基础。

1.1 MVC 思想概述

进入 Struts 2 世界固然令人兴奋,但是 Struts 2 并不是一门完全独立的技术,而是建立在其他 Web 技术之上的一个 MVC 框架。所以,要掌握好 Struts 2 这门技术,一定要理解 MVC 设计思想。那么什么是 MVC 设计思想呢?

在我们实际的开发过程中,随着应用系统的逐渐增大,如果依然把所有的处理逻辑都放在 JSP 页面中,那将是一场噩梦。而采用 MVC 设计模式进行开发将会避免这一噩梦。

MVC 设计思想就是将应用中各组件按功能进行分类,不同的组件使用不同的技术来实现,严格点,还可以按层进行划分,不同组件被严格限制在其所在层内,各层通过松耦合的方式组织在一起,从而使应用程序有更好的扩展性。

1.1.1 传统的 Model 1 和 Model 2

Java Web 应用的发展经历了 Model 1 和 Model 2 两个时代,从 Model 1 发展到 Model 2 是历史的进步,更是众多程序设计开发人员心血的结晶。

在 Model 1 模式下,整个 Web 应用几乎全部由 JSP 页面组成,JSP 页面接收处理客户端发送过来的请求,请求处理完毕直接做出响应。Model 1 阶段使用 JavaBean 来处理数据库的连接和访问操作,如图 1.1 所示。

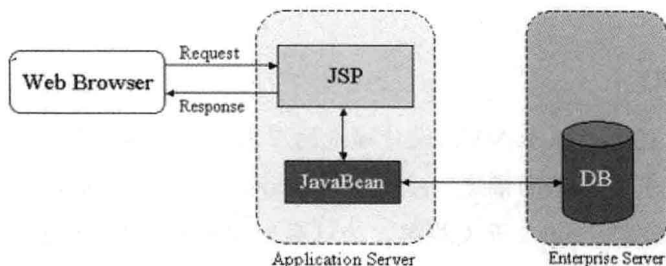


图 1.1 Model 1 体系结构

Model 1 模式的实现比较简单,适用于快速开发小规模的项目。但是使用 Model 1 有一定的缺点:JSP 页面身兼视图(View)和控制器(Controller)两种角色,将控制逻辑和表现视图混杂在一起,从而导致代码的重用性非常低,增加了应用的扩展和维护的难度。

Model 2 已经是基于 MVC 的设计模式。在 Model 2 中 Servlet 作为控制器角色,负责接收客户端发送的请求,在 Servlet 中只包含控制逻辑和前端处理;然后,由 Servlet 负责调用 JavaBean 完成逻辑处理,最后由 Servlet 转发相应的处理结果到 JSP 页面进行显示。Model 2 的具体实现方式如图 1.2 所示。

由图 1.2 可见,在 Model 2 中,JSP 不再承担控制器的角色,它仅仅是一个表现层,仅用于将结果呈现给用户,JSP 页面的请求与 Servlet 进行交互,而 Servlet 负责与后台的 JavaBean 通信。在 Model 2 中,模型(Model)由 JavaBean 充当,视图(View)由 JSP 页面充当,而控制器(Controller)由 Servlet 充当。

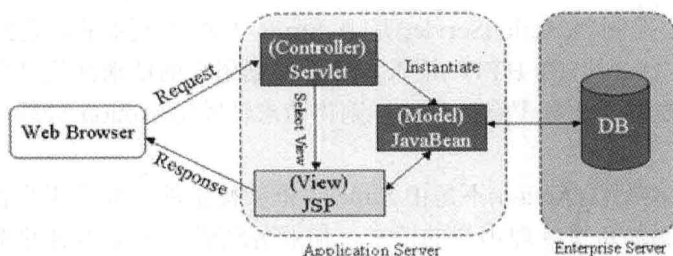


图 1.2 Model 2 体系结构

在 Model 2 中，由于引入了 MVC 设计模式，使 Model 2 具有组件化的特点，更适用于大型项目的开发，但同时也增加了应用开发的复杂程度。原本需要一个简单的 JSP 页面就能实现的功能，现在需要拆分为多个部分，通过协同工作来实现这一功能，就要求我们学习者花费更多的时间，才能真正掌握其设计和实现过程。

1.1.2 MVC 思想及特点

MVC 并不是 Java 语言所特有的设计思想，也不是 Web 技术所特有的思想，在所有面向对象程序设计语言中都可以采用这一思想进行应用程序的开发。

MVC 思想将一个应用分成三个基本部分：Model(模型)、View(视图)和 Controller(控制器)，让这三个部分以最低的耦合进行协同工作，从而提高应用的可扩展性及可维护性。

概括起来，MVC 有如下特点：

- 一个模型对应多个视图，提高了代码的可重用性，模型的变化易于维护。
- 模型返回的数据与视图分离，模型返回的数据可以应用任何的显示技术。
- 应用分割为三层，降低了各层之间的耦合，提高了应用的可扩展性。

1.1.3 常见 MVC 技术

常见 MVC 技术如下。

- Struts：是最经典的 MVC 框架，是 Apache Jakarta 项目组的一个开源项目。
- Struts 2：是 Apache Struts 和 WebWork 组合产生的新产品，是最灵活、最简单的 MVC 组件。
- WebWork：是致力于组件化和代码重用的 J2EE Web 框架，Struts 2 的前身。
- SpringMVC：是 Spring FrameWork 整合 Spring 优势推出的 MVC 组件。
- JSF：是构建 Web 应用程序的一种标准框架。

1.2 Struts 1 概述

在过去，Struts 1 无论从市场占有率上，还是从拥有的开发人群上，都可以说是 MVC 框架中不容辩驳的胜利者。对于 Struts 1 框架而言，它的核心就是控制器，Struts 1 的控制器由两部分组成：核心控制器和业务逻辑控制器。

Struts 1 的核心控制器(ActionServlet)是由 Struts 1 框架提供的。它继承自 HttpServlet 类,该控制器负责拦截所有的 HTTP 请求,然后根据用户的请求决定是否需要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。

Struts 1 的业务控制器(Action)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的请求,但业务逻辑控制器本身并不具有处理能力,而是调用 Model 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 MVC 框架来说,其实大部分只实现了控制器部分,由它负责调用业务逻辑组件,并负责控制器与视图技术的交互。

当客户端向服务器端发送请求时,请求首先被 Struts 1 的核心控制器 ActionServlet 拦截,ActionServlet 然后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 本身并不具备处理能力,只是进行业务逻辑控制,它还需要调用相应的模型来完成处理。当处理完毕客户的请求后,Action 就将处理结果通过视图 JSP 页面呈现给用户,如图 1.3 所示。

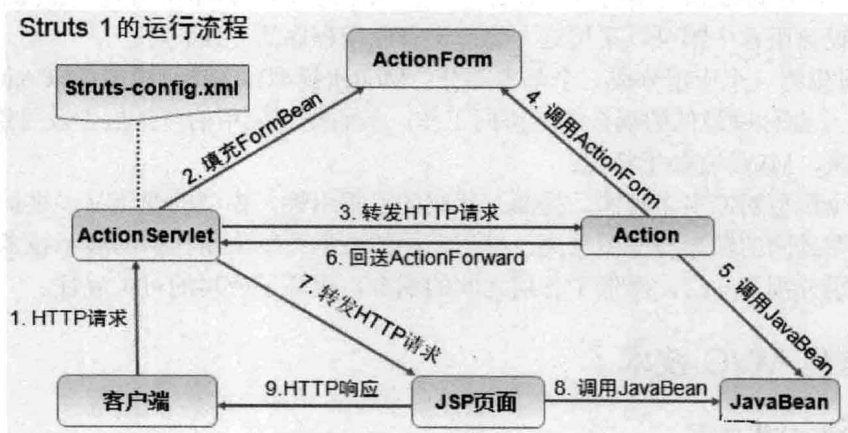


图 1.3 Struts 1 的运行流程

随着 Web 技术的不断发展和 Web 应用的不断扩大, Struts 1.0 的诸多缺陷和局限性也逐渐表现了出来,例如以下几点。

(1) Struts 1 所支持的表现层技术比较单一

Struts 1 只支持 JSP 作为表现层技术,没有提供目前流行的 FreeMarker, Velocity 等表现层技术的支持。

(2) Struts 1 与 Servlet API 耦合严重,难于测试

在 Struts 1 中的业务逻辑控制器内充满了大量的 Servlet API 的应用,而 Servlet API 通常是由 Web 容器来负责实例化的,所以 Struts 1 严重依赖于 Web 服务器,一旦脱离了 Web 服务器,Action 则无法进行单元测试。如下面的 Action 代码片段所示:

```
//在 Struts 1 中业务逻辑控制器必须继承 Struts 1 提供的 Action 类
public class LoginAction extends Action {
    //execute 用来处理用户的请求
```

```
public ActionForward execute(ActionMapping mapping, ActionForm form,
    HttpServletRequest request, HttpServletResponse response)
    throws Exception {
    //在 Struts 1 中会把用户请求的数据封装到 ActionForm 中
    //然后再把 ActionForm 对象强制转换为用户登录的 LoginForm
    LoginForm loginForm = (LoginForm)form;
    //再从 LoginForm 中获取用户请求的数据, 进行相关判断
    if("admin".equals(loginForm.getName())
        && "123456".equals(loginForm.getPassword())) {
        //用户名和密码正确, 返回一个 ActionForward 对象
        return mapping.findForward("success");
    } else {
        //用户名或密码有误, 返回一个 ActionForward 对象
        return mapping.findForward("fail");
    }
}
```

假如们要测试上面的 Action 类的 execute 方法, 该方法有 4 个参数: ActionMapping、ActionForm、HttpServletRequest 和 HttpServletResponse, 初始化这 4 个参数比较困难, 尤其是 HttpServletRequest 和 HttpServletResponse 两个参数, 这两个参数通常都是由 Web 容器负责实例化的, 因为 HttpServletRequest 和 HttpServletResponse 两个参数是 Servlet API, 严重依赖于 Web 容器。所以, 一旦脱离了 Web 容器, Action 进行单元测试将变得非常困难。从 2008 年年底到 2013 年年初, Struts 1 没有发布更新的版本。

1.3 Struts 2 概述

Struts 2 虽然是在 Struts 1 的基础上发展起来的, 但它实质上是以 WebWork 为核心。Struts 2 并没有继承 Struts 1 的设计理念, 而是使用了 WebWork 的设计理念。Struts 2 与 Struts 1 相比有很多革命性的改进, 差别也很大, 所以接下来简单介绍一下 WebWork。

WebWork 也是一种基于 MVC 设计模式的开源框架, 它是由 OpenSymphony 组织开发的, 致力于研发组件化和代码重用的 J2EE Web 框架。它对 Struts 2 框架进行了革命性的改进, 引进了很多新的设计思想、概念和功能。相对于 Struts 1 的那些缺陷和局限性, WebWork 是更加优秀的。

- (1) WebWork 支持更多的表现层技术, 如 JSP、FreeMarker、Velocity 等技术。
- (2) WebWork 采用的是一种更加松耦合的设计, 使 Action 不再与任何 Servlet API 耦合。WebWork 的 Action 更像是一个普通的 Java 对象, 这也使得单元测试更加方便。
- (3) WebWork 的 Action 中没有出现任何 WebWork API, 提高了代码的可重用性。

鉴于 WebWork 的优点, 所以 Struts 2 中引入了 WebWork 的设计理念。Struts 2 是对 Struts 1 和 WebWork 两大框架的整合, 使原来使用 Struts 1 和 WebWork 的开发人员都能够很快过渡到使用 Struts 2 框架进行开发。Struts 1、WebWork 和 Struts 2 三者之间的关系如图 1.4 所示。

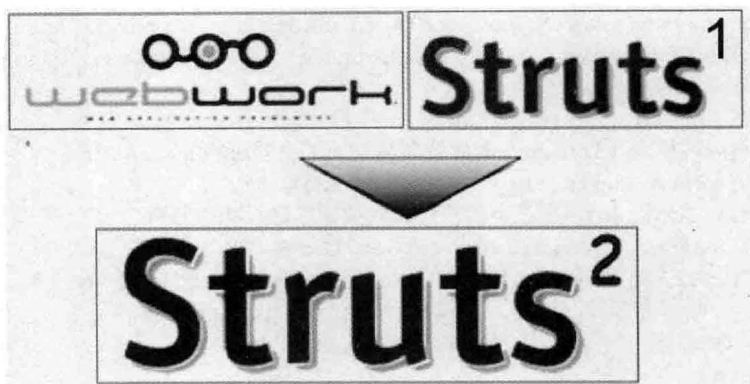


图 1.4 Struts 1、WebWork、Struts 2 的关系

1.3.1 Struts 2 的优势

通过以上对 Struts 2 的介绍不难看出, Struts 2 因为采用了 WebWork 的设计理念, 已经有了很大的改进。

相对于 Struts 1 框架中的 Action 出现了大量的 Servlet API 而言, Struts 2 的 Action 更像一个普通的 Java 类, 该控制器代码中没有耦合任何 Servlet API, 如下面 Struts 2 的 Action 代码所示:

```
public class LoginAction implements Action {

    //提供相应的属性, 用来封装用户请求的数据
    private String name;
    private String password;

    //此处省略 name 的 setter、getter 方法
    ...
    //此处省略 password 的 setter、getter 方法
    ...
    //处理用户请求的 execute 方法
    public String execute() throws Exception {
        if("admin".equals(getName()) && "123456".equals(getPassword())) {
            //用户名和密码正确, 返回一个 String 对象
            return "success";
        } else {
            //用户名或密码有误, 返回一个 String 对象
            return "fail";
        }
    }
}
```

在上面的 Action 代码中, 我们看不到任何的 Servlet API, 当需要处理用户发送过来的 name 和 password 时, Action 并未通过 HttpServletRequest 对象来获取请求参数, 而是直接调用 Action 中的 name 和 password 成员属性, 这两个属性由 Struts 2 中的拦截器负责进行初始化, 以用户请求参数为其赋值。

通过上面的 Action 示例代码不难发现, Struts 1 的 Action 确实太臃肿了, 确实不如 Struts 2 的 Action 优雅、简洁。如果需要对 Struts 2 中 Action 的 execute 方法进行测试, 将非常容易。

关于 Struts 2 更多的优势, 接下来通过表格的形式比较 Struts 2 和 Struts 1 的特性。通过这些特性的比较, 能够清晰地看出 Struts 2 框架的优势和一些新的特性, 如表 1.1 所示。

表 1.1 Struts 2 与 Struts 1 对比

特 性	Struts 1	Struts 2
Action 类	Struts 1 要求 Action 类要扩展自一个抽象基类。Struts 1 的一个共有的问题是面向抽象类编程而不是面向接口编程	Struts 2 的 Action 类实现了一个 Action 接口, 连同其他接口一起来实现可选择和自定义的服务。当然, Struts 2 的 Action 接口并不是必需的, 任何使用 execute() 方法的 POJO 对象都可以被当作 Struts 2 的 Action 对象来使用
线程模型	Struts 1 的 Action 类是单例类, 因为只有一个实例来控制所有的请求。单例类策略造成了一定的限制, 并且给开发者带来额外的烦恼。Action 资源必须是线程安全或者同步	Struts 2 的 Action 对象为每一个请求都实例化对象, 所以没有线程安全的问题
Servlet 依赖	Struts 1 的 Action 类依赖于 Servlet API。Action 被调用时, 以 HttpServletRequest 和 HttpServletResponse 作为参数传给 execute() 方法	Struts 2 的 Action 与容器无关。Servlet 上下文被表现为简单的 Maps, 允许 Action 被独立地测试。Struts 2 的 Action 可以访问最初的请求。但是, 应该尽可能避免其他元素直接访问 HttpServletRequest 或 HttpServletResponse
易测性	测试 Struts 1 的主要问题是 execute() 方法暴露了 Servlet API, 这使得测试要依赖于容器	Struts 2 的 Action 可以通过初始化、设置属性和调用方法来测试。依赖注入的支持也使测试变得更简单
捕获输入	Struts 1 使用 ActionForm 对象来捕获输入。与 Action 一样, 所有的 ActionForm 必须扩展基类。因为其他的 JavaBean 不能作为 ActionForm 使用, 开发者经常创建多余的类来捕获输入	Struts 2 直接使用 Action 属性作为输入属性, 消除了对第二个输入对象的需求。Action 属性能够通过 Web 页面上的 taglibs 访问。Struts 2 也支持 ActionForm 模式, 简化了 taglib 对 POJO 输入对象的引用
表达式语言	Struts 1 整合了 JSTL, 所以使用 JSTL 的表达式语言。表达式语言支持基本的图像对象移动, 但是对集合和索引属性的支持很弱	Struts 2 支持 JSTL, 但是还支持一个更强大和灵活的表达式语言 Object Graph Notation Language (OGNL)

续表

特 性	Struts 1	Struts 2
类型转换	Struts 1 的 ActionForm 属性经常都是 String。Struts 1 使用 Commons-Beanutils 来进行类型转换。转换每一个类，而不是为每一个实例进行配置	Struts 2 内置了转换器，对于日期、整数或浮点数等类型，会自动去调用相应的转换器进行类型转换，为编程带来极大的方便
视图支持	Struts 1 只支持 JSP 作为其表现层技术，没有提供对目前流行的 FreeMarker、Velocity 等表现层的支持	Struts 2 提供了对 FreeMarker、Velocity 等模板技术的支持，并且其配置也很简单

1.3.2 Struts 2 的体系结构

在 Struts 2 中通过拦截器来处理用户的请求，从而允许用户的业务逻辑控制器和 Servlet 分离，在处理请求的过程中以用户的业务逻辑控制器为目标，创建一个控制器代理，控制代理回调业务控制器中的 `execute` 方法来处理用户的请求，该方法的返回值决定了 Struts 2 以怎样的视图资源呈现给用户，如图 1.5 所示。

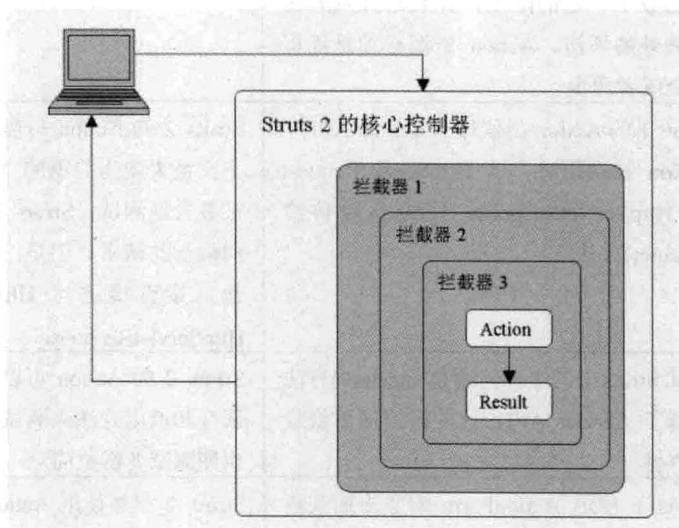


图 1.5 Struts 2 体系结构

从图 1.5 中，我们可以发现 Struts 2 框架的大致处理流程如下。

- (1) 浏览器发送请求，例如请求 `login.action`、`user/test.pdf` 等。
- (2) 核心控制器 `StrutsPrepareAndExecuteFilter` 根据请求决定调用合适的 Action。
- (3) Struts 2 的拦截器链自动对请求进行相关应用的拦截，如 `validation`(数据验证)或文件的上传下载等功能。
- (4) 回调 Action 的 `execute` 方法，该 `execute` 方法先获取用户请求参数，然后执行某种数据库的操作，既可以将数据保存到数据库，也可以从数据库中查询数据。实际上，Action 只是一个控制器，它会调用业务逻辑组件来处理用户的请求。