

普通高校本科计算机专业特色教材精选·算法与程序设计

JavaEE轻量级框架

Struts2+Spring+Hibernate整合开发

史胜辉 王春明 陆培军 编著

清华大学出版社



普通高校本科计算机专业特色教材精选·算法与程序设计

JavaEE轻量级框架

Struts2+Spring+Hibernate整合开发

史胜辉 王春明 陆培军 编著



清华大学出版社
北京

内 容 简 介

本书以较小的篇幅介绍了 Struts 2、Hibernate 3 和 Spring2 三个框架相关的基本知识和相关技术,目的是减少初学者的学习难度,使读者能用最少的时间掌握三个框架的基本使用方法。Struts 2 部分主要内容是 Struts 标签库、Action 和 Struts 2 的配置文件。Hibernate 部分主要讲解 ORM 的概念、HQL 语言及 Hibernate 的基本应用。Spring2 部分主要讲解 IOC 和 AOP 的实现及 Spring2 的基本应用。

本书可作为计算机相关专业本科生的教材,还可作为有一定的 JSP 和 Web 开发基础的计算机编程爱好者的参考书。

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

JavaEE 轻量级框架 Struts2+Spring+Hibernate 整合开发/史胜辉,王春明,陆培军编著.--北京:清华大学出版社,2014

普通高校本科计算机专业特色教材精选·算法与程序设计

ISBN 978-7-302-34894-8

I. ①J… II. ①史… ②王… ③陆… III. ①Java 语言—程序设计—高等学校—教材
IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 311124 号

责任编辑:袁勤勇 王冰飞

封面设计:傅瑞学

责任校对:焦丽丽

责任印制:宋 林

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>, 010-62795954

印 装 者:北京密云胶印厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:14.5

字 数:331千字

版 次:2014年2月第1版

印 次:2014年2月第1次印刷

印 数:1~2000

定 价:29.00元

产品编号:045075-01

前言

PREFACE

在当今的软件开发中 Java 是最热门的编程语言,这是毋庸置疑的。Java Web 开发的高效和便利性是 Java 开发最流行的原因之一。而在 Java Web 开发中的三个框架技术 Struts、Spring 和 Hibernate 有着功不可没的作用。Struts 2+ Spring+ Hibernate(SSH)三个框架具有开发高效、使用灵活的特点,已经得到程序员的认可。在本教材中对三个框架技术作了全面系统的概述,同时更侧重于实践操作,教材中对理论的叙述并不是太多,我们主要是通过一些实例来讲述相关的概念和一些框架技术的使用方法。三个框架在教材中所占的比重也是不同的,其中 Struts 2 和 Hibernate 所占的比重要大一些, Spring 所占的要少一些,这主要是从教学学时受限考虑的。在每章都有一个或两个实验,这样便于教师组织教学。其各章的具体内容如下:

第 1 章主要内容是 Struts 2、Spring、Hibernate 三个框架的介绍。

第 2 章主要内容是 Struts 2 基本工作原理、开发环境、Action 的创建和配置,其中一个比较重要的内容是 Action 的多方法调用技术的使用。

第 3 章主要内容是 OGNL 表达式语言基础知识、常用的标签及使用方法,在这里我们只给出了常用的二十几个标签,其中主要的是 UI 部分标签比较多。

第 4 章主要内容是拦截器的定义和使用、Struts 2 的国际化和 Struts 2 校验框架技术。Struts 2 校验方法有三种,我们这里只介绍了其中的框架校验方法。

第 5 章主要内容包括 ORM 基本概念、Hibernate 体系结构、持久化类、Hibernate 配置文件、关系对象—映射文件、Hibernate API 操作数据库,这一章的最后给出了在 MyEclipse 中创建支持 Hibernate 框架工程,并添加数据库记录的详细步骤。

第 6 章主要包括一对多的双向关联映射和多对多映射。

第 7 章主要包括 HQL 查询和 Criteria 对象查询。

第 8 章是 Struts 2 与 Hibernate 3 集成开发的一个完整案例。

第 9 章主要内容包括 Spring 简介、Spring 框架的系统架构、Spring 框架主要的技术特性、Spring 开发环境。

第 10 章主要内容包括 IoC 的概念及实现方法、AOP 的概念及实现方法、一个完整的 SSH 开发案例。

本书具有以下特色：

(1) 实践性强。教材中对理论内容没有太多的讲解，主要是以案例的方式来讲解一些框架技术的使用方法，书中给出了大量的实践案例。

(2) 容易操作。所有的案例设计遵循简单、实用、便于调试的原则，都提供了经过调试的源代码。

(3) 方便教学。为方便教师组织教学，提供了完整的 PPT 讲义，每一章都有一到两个实验。

本书可作为计算机相关专业本科生的教材，还可作为有一定的 JSP 和 Web 开发基础的计算机编程爱好者的参考书。

本书的编写过程中得到了很多同事的帮助，作者在此深表谢意。由于作者水平有限，书中难免会有一些不足之处，敬请读者批评指正。

作者

2014 年 1 月

目 录

CONTENTS

第 1 章 3 个框架的概述	1
1.1 软件开发中需要解决的问题	1
1.2 Struts 2、Hibernate 和 Spring 框架	2
第 2 章 Struts 2 的基础知识	5
2.1 MVC 模式	5
2.2 Struts 2 的基本工作原理	6
2.2.1 Struts 2 的框架结构	6
2.2.2 Struts 2 的控制流程	7
2.3 Struts 2 的工作环境	8
2.3.1 搭建 Struts 开发环境	8
2.3.2 创建一个简单的 Struts 2 程序	8
2.3.3 Struts 2 的配置	13
2.4 Action 的创建和配置	19
2.4.1 Action 的创建	19
2.4.2 Action 的多方法调用	22
2.4.3 Result 类型	24
2.4.4 在 Action 中调用 Web 对象	26
实验一 创建一个简单的 Struts 2 应用程序	27
实验二 创建一个 Action 多方法调用的 Struts 2 应用程序	27
第 3 章 Struts 2 标签库及使用	29
3.1 Struts 2 标签库概述	29
3.2 OGNL 表达式语言	30
3.2.1 OGNL 基础知识	30
3.2.2 OGNL 的集合操作	33
3.3 Struts 2 标签	37

3.3.1	表单标签	37
3.3.2	控制标签	51
3.3.3	其他常用标签	59
3.4	标签综合应用	66
3.4.1	综合实例 1	66
3.4.2	综合实例 2	67
实验三	Struts 2 标签的应用	70
第 4 章	Struts 2 的高级特性	75
4.1	拦截器	75
4.1.1	Struts 2 中拦截器的工作过程	75
4.1.2	在 Struts 2 中内置拦截器	76
4.1.3	配置拦截器	77
4.1.4	拦截器的使用	78
4.1.5	自定义拦截器	78
4.2	Struts 2 的国际化	82
4.2.1	在 Struts 2 中加载全局资源文件	82
4.2.2	访问国际化消息	82
4.3	Struts 2 校验框架	86
4.3.1	编写校验规则文件	86
4.3.2	Struts 2 的内建校验规则	89
4.4	Struts 2 开发案例	94
实验四	Struts 2 校验框架的应用	97
第 5 章	Hibernate 的基础知识	99
5.1	ORM 基本概念	99
5.1.1	ORM 框架简介	99
5.1.2	ORM 中的映射关系	100
5.2	Hibernate 的体系结构	100
5.3	Hibernate API 简介	102
5.4	Hibernate 的配置文件	103
5.5	Hibernate 中的持久化类	105
5.6	Hibernate 的对象——关系映射文件	107
5.7	通过 Hibernate API 操纵数据库	109
5.8	在 MyEclipse 中使用 Hibernate	114
5.9	Session 的缓存	119
实验五	完成表的查询和删除操作	121

第 6 章 Hibernate 的关联映射	123
6.1 一对多关联	123
6.1.1 单向多对一关联	123
6.1.2 双向一对多关联	126
6.2 多对多关联	130
实验六 完成一对多关联操作	135
第 7 章 Hibernate 查询	137
7.1 Hibernate 检索策略	137
7.1.1 Hibernate 检索策略简介	137
7.1.2 get()方法与 load()方法的区别	141
7.2 Hibernate 检索方式	142
7.3 HQL 查询	142
7.3.1 from 子句	144
7.3.2 select 子句	144
7.3.3 where 子句	145
7.3.4 聚集函数	145
7.4 Criteria 查询	146
实验七 完成表的查询操作	147
第 8 章 Struts 2 与 Hibernate 3 的集成开发案例——网上书店	149
8.1 数据库的设计	149
8.2 创建项目工程	151
8.3 编写数据库操作类	151
8.4 编写 JSP 文件	154
8.5 编写 Action 类	156
8.6 校验框架	160
8.7 Struts 配置文件	162
实验八 查询某一用户的所有订单	163
第 9 章 Spring 的基础知识	165
9.1 Spring 简介	165
9.2 Spring 框架的系统架构	167
9.3 Spring 框架的主要技术特性	169
9.4 Spring 开发环境	170
9.5 Spring 程序示例	174
实验九 Spring 的 IoC 实现案例	176

第 10 章 Spring 的基本用法	181
10.1 控制反转的实现	181
10.1.1 什么是 IoC	181
10.1.2 IoC 配置文件示例	183
10.1.3 Spring 框架中的依赖注入技术	184
10.1.4 依赖注入的 3 种形式	185
10.2 BeanFactory 和 ApplicationContext 的工作原理	188
10.2.1 BeanFactory 介绍	188
10.2.2 ApplicationContext 介绍	190
10.2.3 容器中 Bean 的生命周期	190
10.2.4 ApplicationContext 中 Bean 的生命周期	193
10.3 面向切面编程	198
10.3.1 面向切面编程的概念	199
10.3.2 面向切面编程的实现	200
10.4 Spring 与 Struts 2 的整合	205
10.5 Spring 与 Hibernate 的整合	207
10.6 Struts 2+Hibernate+Spring 的整合	211
实验十 SSH 开发案例	223

3 个框架的概述

典型的 J2EE(即 Java EE)三层结构分为表现层、中间层(业务逻辑层)和数据服务层,三层体系将业务规则、数据访问及合法性校验等工作放在中间层处理,客户端不直接与数据库交互,而是通过组件与中间层建立连接,再由中间层与数据库交互。

1.1 软件开发中需要解决的问题

在软件开发过程中通常要解决开发效率、需求变更、技术更新等问题,下面笔者就这几个问题谈一下自己的看法。

软件工程是个特殊行业,不同于传统工业(如电器、建筑及汽车等行业)。这些行业的产品一旦开发出来,交付用户使用后很少需要后续维护。但软件行业不同,软件产品的后期运行维护是个巨大的工程,单纯从前期开发时间上考虑其开发效率是不理智的,也是不公平的。众所周知,对于传统的 JSP 技术,将整个站点的业务逻辑和表现逻辑都混杂在 JSP 页面里,从而导致页面的可读性相当差,可维护性非常低。即使需要简单改变页面的按钮,也不得不冒着破坏系统的风险打开页面文件。但采用严格分层 J2EE 架构,则可完全避免这个问题。对表现层的修改即使发生错误,也绝对不会将错误扩展到业务逻辑层,更不会影响持久层。因此,采用 J2EE 分层架构,即使前期的开发效率稍微低一点也是值得的。

在软件开发过程中,很少有软件产品的需求从一开始就完全是固定的。客户对软件的需求,是随着软件开发过程的深入不断明晰起来的。因此,开发人员经常会遇到软件开发到一定程度时,由于客户对软件需求发生了变化,使得软件的实现不得不随之改变的情况。当软件实现需要改变时,是否可以尽可能多地保留已开发的程序,尽可能少地改变软件的实现,从而满足客户需求的变更呢?我们的理想是有一种优秀的解耦架构能满足这种需求,这种架构就是 J2EE 的分层架构。在优秀的分层架构里,控制层依赖于业务逻辑层,但绝不与任何具体的业务逻辑组件耦合,而只与接口

耦合;同样,业务逻辑层依赖于 DAO 层,也不会与任何具体的 DAO 组件耦合,而是面向接口编程。采用这种方式的软件实现,即使软件的部分发生改变,其他部分也尽可能不要改变。

针对软件开发中遇到的问题,我们可以参考传统的硬件行业。在计算机硬件中有大量的接口规范,例如 PCI 接口、显卡或者网卡,只要其遵守 PCI 的规范,就可以插入主板,与主板通信。至于这块卡内部的实现,不是主板所关心的,这也正是面向接口编程的好处。假如需要提高计算机的性能,需要更新显卡,只要更换另一块 PCI 接口的显卡即可,而不是将整台计算机换掉。如果一台计算机不是采用各种接口组合在一起,而是做成整块,那将意味着即使只需要更新网卡,也要换掉整台计算机。同样,对于软件中的各组件,当一个组件需要重构时,尽量不要影响其他组件。实际上,这是最理想的情况,即使采用目前最优秀的架构,也会有或多或少的影响,这也是软件工程需要努力提高的地方。

软件行业的技术更新很快,虽然软件行业的发展不快,但小范围的技术更新特别快。一旦由于客观环境的变化不得不更换技术时,如何保证系统的改变最小呢?答案是选择优秀的架构。

1.2 Struts 2、Hibernate 和 Spring 框架

Struts 的第一个版本是在 2001 年 5 月发布的。它的最初设想是通过结合 JSP 和 Servlet,使 Web 应用的视图和业务/应用逻辑得以清晰地分离开来。在 Struts 之前,最常见的做法是在 JSP 中加入业务和应用逻辑,或者在 Servlet 中通过 `println()` 来生成视图。

自从第一版发布以来,Struts 实际上已经成为业界公认的 Web 应用标准。它的炙手可热也为自己带来了改进和变更,所以不仅要跟上 Web 应用框架不断变化的需求,还要与日益增多、竞争激烈的众多框架的特性相融合。到最后,产生了几个下一代 Struts 的解决方案,其中最受瞩目的两个方案是 Shale 和 Struts Ti。Shale 是一个基于构件的框架,并在最近成为 Apache 的顶级项目,而 Struts Ti 则是在 Struts 的成功经验基础上继续坚持对前端控制器(Front Controller)和 MVC(Model-View-Controller)模式进行改进。

WebWork 项目是在 2002 年 3 月发布的,它对 Struts 式框架进行了革命性改进,引进了很多新的思想、概念和功能,但和原 Struts 代码并不兼容。WebWork 是一个成熟的框架,经过了几次重大的改进与发布。在 2005 年 12 月,WebWork 与 Struts Ti 宣布合并。与此同时,Struts Ti 改名为 Struts Action Framework 2.0,成为 Struts 真正的继承者。

要注意的是,合并和改名并不是说 Struts 或 WebWork 项目已经停止开发了。其实,人们对这两个项目的兴趣仍然很高,而且有很多开发者仍然愿意使用它们,因此这两个项目还在继续开发,继续修复 Bug,继续改进功能和添加新功能。

在数据持久层,我们通常使用 JDBC 连接来读写数据库,最常见的就是打开数据库连接、使用复杂的 SQL 语句进行读写、关闭连接,获得的数据又需要在转换或封装后向外传,这是一个非常烦琐的过程。

这时出现了 Hibernate 框架,它需要用户创建一系列持久化类,每个类的属性都可以简单地看做和一张数据库表的属性一一对应,当然,也可以实现关系数据库中的各种关系表与类之间的一一对应。这样,当我们需要相关操作时,不必再关注数据库表。不用再去一行行地查询数据库,只需要持久化类就可以完成增、删、改、查的功能。这使得我们的软件开发真正面向对象,而不是面向混乱的代码。笔者的感受是,使用 Hibernate 比使用 JDBC 方式约减少了 80% 的编程量。

现在我们有二个层了,但是每层之间的调用是怎样的呢?例如显示层的 Struts 需要调用一个业务类就需要新建一个对象,然后业务层需要调用持久层的类也需要创建一个对象,通过这种创建对象的方式互相调用就是软件开发中设计最糟糕的体现。简单地说,就是调用者依赖被调用者,它们之间形成了强耦合,如果想在其他地方复用某个类,则这个类依赖的其他类也需要全部包含进来。这样程序就变得很乱,每个类互相依赖、互相调用,复用度极低。如果一个类做了修改,则依赖它的很多类都会受到牵连。为此,出现了 Spring 框架。

Spring 的作用就是完全解耦类之间的依赖关系,一个类如果要依赖什么,那么就是一个接口。至于如何实现这个接口,其实不重要,只要用户拿到一个实现了这个接口的类,就可以轻松地通过 XML 配置文件把实现类注射到调用接口的那个类里,所有类之间的这种依赖关系完全通过配置文件的方式来解决。所以, Spring 框架最核心的内容就是所谓的依赖注射和控制反转。

现在流行的结构是, Struts 负责显示层, Hibernate 负责持久层, Spring 负责中间的业务层,这个结构是目前国内外最流行的 Java Web 应用程序架构。另外,由于 Spring 使用依赖注射以及 AOP(面向方面编程),所以它的内部模式非常优秀,以至于 Spring 自己实现了一个使用依赖注射的 MVC 框架,叫做 Spring MVC。并且,为了很好地处理事务, Spring 集成了 Hibernate,使事务管理从 Hibernate 的持久层提升到了业务层,使用更加方便。

第 2 章

CHAPTER

Struts 2 的基础知识

2.1 MVC 模式

随着应用系统的逐渐增大,系统的业务逻辑复杂度以几何级数方式增加。在这种情况下,如果依然把所有的处理逻辑放在 JSP 页面中,那将是一场噩梦,无论我们进行什么样的改变,都必须打开那些杂乱的 JSP 脚本进行修改。

MVC 将应用中的各组件按功能进行分类,不同的组件使用不同的技术,相同的组件被严格地限制在其所在层内,各层之间以松耦合的方式组织在一起,从而提供良好的封装。Java Web 应用的结构经历了 Model1 和 Model2 两个时代。

在 Model1 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请求,对请求处理后直接做出响应,用少量的 JavaBean 来处理数据库连接、数据库访问等操作,如图 2-1 所示。

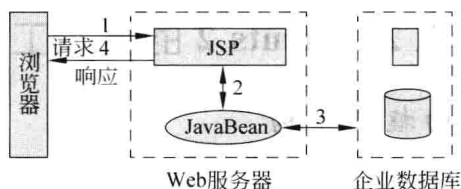


图 2-1 Model1 的流程图

Model1 模式的实现比较简单,适用于快速开发小规模项目。但从工程化的角度看,它的局限性非常明显,JSP 页面身兼 View 和 Controller 两种角色,将控制逻辑和表现逻辑混杂在一起,从而导致代码的重用性非常低,增加了应用的扩展性和维护的难度。

Model2 已经是基于 MVC 架构的设计模式。在 Model2 架构中,Servlet 作为前端控制器,负责接收客户端发送的请求,在 Servlet 中只包含控制逻辑和简单的前端处理;然后,调用后端 JavaBean 来完成实际的逻辑

处理;最后,转发到相应的 JSP 页面处理显示逻辑。其具体的实现方式如图 2-2 所示。

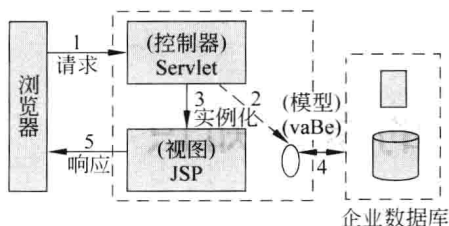


图 2-2 Model2 的程序流程图

正如图 2-2 中所示,在 Model2 模式下 JSP 不再是控制器,它只承担表现层的任务,用于将结果呈现给用户,JSP 页面的请求与 Servlet(控制器)交互,而 Servlet 负责与后台的 JavaBean 通信。在 Model2 模式下,模型(Model)由 JavaBean 充当,视图(View)由 JSP 页面充当,而控制器(Controller)则由 Servlet 充当。

由于引入了 MVC 模式,使 Model2 具有了组件化的特点,适用于大规模应用的开发,但也增加了应用开发的复杂程度。原本需要一个简单的 JSP 页面就能实现的应用,在 Model 2 中被分解成多个协同工作的部分,用户需要花费更多时间才能真正地掌握其设计和实现过程。

MVC 的优点如下:

(1) 多个视图可以对应一个模型,按 MVC 设计模式,一个模型对应多个视图,可以减少代码的复制及代码的维护量,即使模型发生改变,也易于维护。

(2) 模型返回的数据与显示逻辑分离,模型数据可以应用任何的显示技术,例如使用 JSP 页面、Velocity 模板或者直接产生 Excel 文档等。

(3) 应用被分为三层,降低了各层之间的耦合,提供了应用的可扩展性。

(4) 控制层的概念很有效,由于它把不同的模型和不同的视图组合在一起完成不同的请求,因此,可以说控制层包含了用户请求权限的概念。

(5) MVC 更符合软件工程化管理的精神,不同层各司其职,每一层的组件具有相同的特征,有利于通过工程化和工具化产生程序代码。

正是基于 MVC 以上的优点,才有了 Struts 2 框架,Struts 2 是 MVC 的最好实现。

2.2 Struts 2 的基本工作原理

2.2.1 Struts 2 的框架结构

Struts 2 框架按照模块来划分,可以分为 Servlet Filters、Struts 核心模块、拦截器和用户实现部分。Struts 2 的框架结构如图 2-3 所示。

Struts 2 框架中各部分的功能处理如下:

(1) FilterDispatcher 是控制器的核心,是 MVC 的 Struts 2 实现中控制层(Controller)的核心。用户从客户端提交 HttpServletRequest 请求,请求经过 ActionContextCleanUp,再通过其他过滤器 Other Filters、SiteMesh 等到达 FilterDispatcher。

(2) FilterDispatcher 将请求发送给 ActionMapper,ActionMapper 决定需要调用某个 Action,FilterDispatcher 则把请求的处理交给 ActionProxy。

(3) ActionProxy 通过 Configuration Manager(struts.xml)询问框架的配置文件,找到需要调用的 Action 类。例如,用户注册示例将找到 UserReg 类。

(4) ActionProxy 创建一个 ActionInvocation 实例,同时 ActionInvocation 通过代理

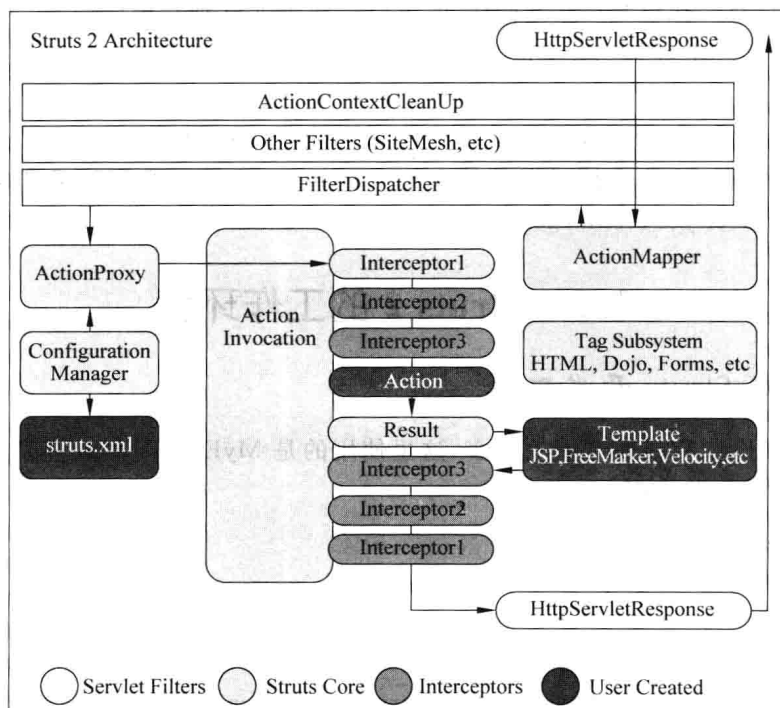


图 2-3 Struts 2 的框架结构

模式调用 Action。但在调用之前, ActionInvocation 会根据配置加载 Action 相关的所有 Interceptor(拦截器)。

(5) 一旦 Action 执行完毕, ActionInvocation 负责根据 struts.xml 中的配置找到对应的返回结果 Result。

Struts 2 的核心控制器是 FilterDispatcher, 有 destroy()、doFilter() 和 init() 3 个重要的方法, 其中, 被经常调用的方法是 doFilter()。在 doFilter() 方法中, 将调用 dispatcher.serviceAction, 该方法如果找到相应的 Action, 将把用户请求交给 ActionProxy。serviceAction() 代码在 Dispatcher.java 中。Struts 2 用于处理用户请求的 Action 实例, 并不是用户实现的业务控制器, 而是 Action 代理。

2.2.2 Struts 2 的控制流程

Struts 2 的控制流程如图 2-4 所示。

Struts 2 的工作流程如下:

- (1) 客户端发送请求, 例如“/regist.action;”。
- (2) 核心控制器 FilterDispatcher 根据请求决定调用合适的 Action。
- (3) WebWork 的拦截器链自动请求应用通用功能, 例如验证、工作流或文件上传等功能。

(4) 回调 Action 的 execute 方法, 该方法先获得

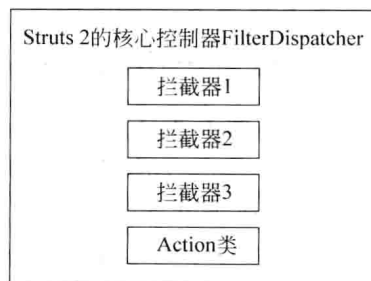


图 2-4 Struts 2 的控制流程

用户请求参数,然后执行某种业务操作,既可以将数据保存到数据库中,也可以从数据库中检索信息。实际上,因为 Action 只是一个控制器,它会调用业务逻辑组件(Model)来处理用户的请求。

(5) Action 的 execute 方法处理结果信息将被输出到浏览器中,可以是 HTML 页面、图像,也可以是 PDF 文档或者其他文档。Struts 2 支持的视图技术非常多,既支持 JSP,也支持 Velocity、FreeMarker 等模板技术。

2.3 Struts 2 的工作环境

2.3.1 搭建 Struts 开发环境

用于开发 Java Web 的工具有很多,这里使用的是 MyEclipse 集成开发环境,下面给出 MyEclipse 的安装步骤。

(1) 安装 MyEclipse 6.5。

(2) 安装配置 Tomcat 6.0。

① 下载 apache-tomcat-6.0.18。

② 启动 MyEclipse 6.5,单击 Window→Preferences 命令,弹出 Preferences 对话框,选择 MyEclipse Enterprise Workbench→Servers→Tomcat→Tomcat 6. x 选项,此时的设置界面如图 2-5 所示。

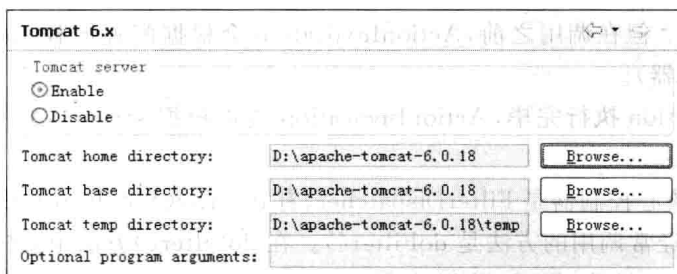


图 2-5 Tomcat 6 服务器

在图 2-5 中选择 Enable,单击 Tomcat home directory 后面的 Browse 按钮,在弹出的对话框中选择 Tomcat 的解压文件夹,设置完成后,单击 OK 按钮。

③ 下载 Struts-2.2.3。

2.3.2 创建一个简单的 Struts 2 程序

有了好的开发工具,开发 Struts 2 应用程序将是一件非常轻松的事,下面以 HelloWorld 程序为例,给出利用 Struts 2 开发 Web 应用程序的详细步骤。

(1) 启动 MyEclipse,新建一个 Web 项目 Struts_HelloWorld,如图 2-6 所示。

(2) 将 Struts 2 的相关 JAR 文件粘贴到 Web 项目的 lib 目录中,如图 2-7 所示。

(3) 在 src 目录下建立一个名为 struts.xml 的配置文件,如图 2-8 所示。