



工业和信息化
人才培养规划教材

Industry And Information
Technology Training
Planning Materials

高职高专计算机系列

Java EE 轻量级框架 应用开发教程

Java EE Application
Development Tutorial

李唯 程永恒 © 主编

李文蕙 张新华 肖英 © 副主编

王路群 © 主审

+ 详解 **SSH 框架开发技术**

+ **丰富**实例，易于理解

+ 重点培养**解决问题能力**



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS



工业和信息化
人才培养规划教材

Industry And Information
Technology Training
Planning Materials

Java EE 轻量级框架 应用开发教程

Java EE Application
Development Tutorial

李唯 程永恒 © 主编

李文蕙 张新华 肖英 © 副主编

王路群 © 主审

人民邮电出版社
北京

图书在版编目 (CIP) 数据

Java EE轻量级框架应用开发教程 / 李唯, 程永恒主
编. — 北京: 人民邮电出版社, 2016.2
工业和信息化人才培养规划教材. 高职高专计算机系
列

ISBN 978-7-115-38917-6

I. ①J… II. ①李… ②程… III. ①JAVA语言—程序
设计—高等职业教育—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2015)第194100号

内 容 提 要

本书共分11章,以MyEclipse为开发平台,结合Tomcat Web服务器、MySQL数据库系统详细讲解SSH(Struts 2+Spring+Hibernate)轻量级框架技术。全书从三个轻量级框架的基础知识入手,对每个框架的应用及整合应用做详细的理论讲解,并结合Struts 2框架应用、Struts 2+Hibernate应用以及SSH整合应用实例,做到理论联系实际。

本书内容全面,案例丰富,既有简单易懂的程序片段、配置代码,也有实际可用的应用系统实例,有利于读者迅速掌握三大框架开发的核心技术。

本书主要面向有Java语言基础,在了解了JSP、Servlet等基础知识后的高校学生、工程技术人员和相关技术的爱好者,既可作为大专院校Java开发类课程的教材,也可以作为Java框架开发人员的入门书籍和参考书。

◆ 主 编 李 唯 程永恒
副 主 编 李文蕙 张新华 肖英
主 审 王路群
责任编辑 范博涛
责任印制 杨林杰

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鑫正大印刷有限公司印刷

◆ 开本: 787×1092 1/16
印张: 15 2016年2月第1版
字数: 382千字 2016年2月北京第1次印刷

定价: 36.00元

读者服务热线: (010)81055256 印装质量热线: (010)81055316
盗版版热线: (010)81055315

前 言 PREFACE

Java 是热门的编程语言,随着互联网的普及推广,Web 应用开发得到了迅速发展,Java Web 技术因其高效性和便利性,得到了广泛的应用。Java EE 是 Sun 公司(后被甲骨文公司收购)为企业级应用推出的标准平台,它提供 Web 服务、组件模型等技术,可以实现企业级的面向服务体系结构和 Web 2.0 应用程序,非常适合 Java Web 开发,是目前 B/S 架构应用中最为成熟的技术之一。仅仅采用 Java EE 中的 JSP 技术进行 Java Web 程序开发,开发时间长,安全性和稳定性也得不到保证,而一个好的框架可以帮助开发人员在短期内搭建结构清晰、可复用性好、维护方便的 Web 应用程序,因此越来越多的项目采用 Java EE 中的框架技术进行开发。SSH (Struts 2+Spring+Hibernate) 是目前流行的一种 Java Web 应用程序轻量级的开发框架组合,表示层用 Struts2,业务层用 Spring,持久层用 Hibernate。SSH 轻量级框架组合并没有和 JavaEE 绑定,启动成本低,降低侵入性,实现了效果极佳的架构重用,在目前的软件开发中得到了广泛的应用。

编者在多年的 Java EE 应用系统开发和教学实践经验的基础上,针对职业教育的特点,根据实际软件开发中对 SSH 框架开发知识的需要组织内容,通过大量示例介绍使用 SSH 轻量级框架开发的知识与技能,使读者学会在 MyEclipse 环境中开发基于 SSH 轻量级框架的 Java EE 程序。

本书力图使用简短的内容介绍基础知识,把重点放在使用 MyEclipse 平台进行 SSH 框架开发的技术,书中附有大量的图表、程序,使读者能正确、直观地理解问题。

全书共分为 11 章,第 1 章介绍 Struts 2 基础知识,包括配置 Struts 2 运行环境、Struts 2 框架工作原理、配置 Struts 2、Action 和类型转换。第 2 章介绍了 Struts 2 标签库,包括标签库概述、使用控制标签、使用数据标签、使用表单 UI 标签。第 3 章介绍了 Struts 2 的高级特性,包括拦截器简介、拦截器配置、自定义拦截器、Struts 2 的国际化、Struts 2 输入校验、应用中的异常处理。第 4 章是一个基于 Struts 2 的注册管理系统项目实训,从项目需求说明、项目系统分析、系统设计与实现等方面进行介绍。第 5 章介绍了 Hibernate 的基础知识,包括持久化概述、ORM 基本概念、Hibernate 的体系结构、Hibernate 执行流程、使用 MyEclipse 工具进行 Hibernate 开发、Hibernate 进行数据库操作。第 6 章介绍了 Hibernate 的关联映射,包括多对一映射、一对多映射、一对一映射、多对多映射以及双向映射。第 7 章介绍了 Hibernate 查询,包括 Hibernate 的检索方式、HQL 检索方式、条件查询和 SQL 检索方式。第 8 章是一个 Struts 2 与 Hibernate 3 的集成开发案例,从项目需求说明、数据库的设计、系统详细设计和系统实现等方面进行了介绍。第 9 章介绍了 Spring 的基础知识,包括 Spring 简介、Spring 框架的系统架构、Spring 框架的主要技术特性和 Spring 开发环境。第 10 章介绍了 Spring 的用法,包括控制反转的实现、面向切面编程以及 Spring 与 Struts 2、Hibernate 进行整合。第 11 章是一个基于 Struts 2+Hibernate+ Spring 的网上书城项目实训,从项目需求说明、数据库分析和网上书城设计与实现等方面进行了说明。

本书由武汉软件工程职业学院李唯、程永恒担任主编,李文蕙、张新华、肖英担任副主编,王路群教授担任主审。其中第 1~4 章由程永恒、李文蕙编写,第 5~8 章、第 11 章由李唯、肖英编写,第 9、10 章由张新华编写。参与本书的编写工作的还有武汉软件工程

职业学院董宁、刘洁、肖奎、刘嵩、谢日星、鄢军霞、马力、罗炜、赵丙秀，以及一些企业人员。全书由李唯和程永恒拟定编写大纲，由李唯负责统编全稿。对在编写过程中提供了帮助和支持的同事和朋友，在此表示衷心的感谢。

本书得到了来自湖北省教育科学规划 2015 年度重点课题（2015GA076）的资助，也是该课题的阶段研究成果。”

由于编者水平有限，书中不妥或错误之处在所难免，恳请广大读者批评指正。

编者

2015 年 11 月

目 录 CONTENTS

第 1 章 Struts 2 基础知识 1

1.1	配置 Struts 2 运行环境	1	1.5.1	实现 Action 控制类	12
1.2	第一个 Struts 2 实例	3	1.5.2	访问和配置 Action	16
1.3	Struts 2 框架工作原理	7	1.5.3	动态方法调用	18
1.4	配置 Struts 2	9	1.5.4	使用 Struts 2 内建类型转换器	20
1.4.1	配置 web.xml	9	1.6	本章小结	23
1.4.2	配置 struts.xml	10	1.7	课内实训	24
1.4.3	配置 struts.properties	12	1.8	实训题	24
1.5	Action 和类型转换	12			

第 2 章 Struts 2 标签库 25

2.1	OGNL	25	2.4.5	param 标签	41
2.2	标签库概述	29	2.4.6	property 标签	42
2.2.1	使用标签库的优势	29	2.4.7	set/url/date 标签	42
2.2.2	Struts 2 标签库分类	30	2.5	使用表单 UI 标签	45
2.3	使用控制标签	30	2.5.1	表单标签的通用属性	45
2.3.1	if/else if/else 标签	30	2.5.2	form 标签	47
2.3.2	iterator/append/merge/generator/ subset/sort 标签	31	2.5.3	文本域型标签	47
2.4	使用数据标签	37	2.5.4	选择型标签	48
2.4.1	action 标签	37	2.5.5	其他简单标签	52
2.4.2	bean 标签	39	2.5.6	复杂操作标签	52
2.4.3	debug 标签	40	2.6	本章小结	52
2.4.4	include 标签	40	2.7	课内实训	52
			2.8	实训题	53

第 3 章 Struts 2 的高级特性 54

3.1	拦截器简介	54	3.3.1	实现拦截器类	62
3.1.1	拦截器的工作原理	54	3.3.2	使用自定义拦截器	63
3.1.2	拦截器意义	55	3.4	Struts 2 的国际化	66
3.1.3	拦截器角色	56	3.4.1	Struts 2 实现国际化的机制	66
3.2	拦截器配置	56	3.4.2	加载国际化资源文件的方式	69
3.2.1	配置拦截器	56	3.4.3	实现自由选择语言环境	69
3.2.2	使用拦截器	59	3.5	Struts 2 输入校验	72
3.2.3	默认拦截器	60	3.5.1	Struts 2 输入校验的必要性	72
3.3	自定义拦截器	62	3.5.2	Struts 2 手动完成输入校验	72

3.5.3 自定义校验器	74	3.7 本章小结	78
3.6 应用中的异常处理	75	3.8 课内实训	78
3.6.1 Struts 2 异常处理机制	75	3.9 实训题	78
3.6.2 Struts 2 异常处理实例	76		

第 4 章 基于 Struts 2 的注册管理系统项目实训 79

4.1 项目需求说明	79	4.3.2 项目代码实现	83
4.2 项目系统分析	80	4.4 本章小结	107
4.3 系统设计与实现	81	4.5 实训题	107
4.3.1 数据库设计	82		

第 5 章 Hibernate 的基础知识 108

5.1 持久化概述	108	5.6 Hibernate 的数据库操作	118
5.2 ORM 基本概念	108	5.6.1 Hibernate 相关文件	118
5.3 Hibernate 的体系结构	110	5.6.2 Hibernate 的数据库操作	122
5.4 Hibernate 的执行流程	111	5.7 本章小结	127
5.5 使用 MyEclipse 工具进行 Hibernate 开发	113	5.8 课内实训	127
		5.9 实训题	127

第 6 章 Hibernate 的关联映射 128

6.1 多对一映射	128	6.4 多对多映射	140
6.1.1 多对一映射配置	128	6.4.1 多对多映射配置	140
6.1.2 测试多对一映射	131	6.4.2 测试多对多映射	143
6.2 一对多映射	133	6.5 双向映射	145
6.2.1 一对多映射配置	133	6.5.1 一对多双向映射	145
6.2.2 测试一对多映射	135	6.5.2 其他双向映射	148
6.3 一对一映射	137	6.6 本章小结	148
6.3.1 一对一映射配置	137	6.7 课内实训	148
6.3.2 测试一对一映射	140	6.8 实训题	148

第 7 章 Hibernate 查询 150

7.1 Hibernate 的检索方式	150	7.3.2 使用 QBC 查询数据	155
7.2 HQL 检索方式	151	7.4 本地 SQL 检索方式	156
7.2.1 HQL 简介	151	7.5 本章小结	156
7.2.2 使用 HQL 查询数据	151	7.6 课内实训	156
7.3 QBC 检索方式	154	7.7 实训题	157
7.3.1 QBC 检索方式简介	154		

第 8 章 Struts 2 与 Hibernate 3 的集成开发案例 158

8.1 项目需求说明	158	8.3.2 部分顺序图	162
8.2 数据库的设计	159	8.4 系统的实现	163
8.2.1 概念设计	159	8.4.1 系统代码介绍	163
8.2.2 数据库表设计	159	8.4.2 页面效果展示	166
8.3 系统详细设计	161	8.5 本章小结	168
8.3.1 系统用例设计	161	8.6 实训题	168

第 9 章 Spring 的基础知识 169

9.1 Spring 简介	169	9.5 本章小结	177
9.2 Spring 框架的系统架构	171	9.6 课内实训	177
9.3 Spring 框架的主要技术特性	172	9.7 实训题	177
9.4 Spring 开发环境	173		

第 10 章 Spring 的用法 178

10.1 控制反转的实现	178	10.2.2 AOP 实例	193
10.1.1 什么是 IoC	178	10.3 Spring 与 Struts 2 的整合	196
10.1.2 Spring 框架中的依赖注入技术	179	10.4 Spring 与 Hibernate 的整合	200
10.1.3 依赖注入的形式	183	10.5 Struts 2+Spring+Hibernate 的整合	207
10.1.4 Bean 与 Spring 容器	186	10.6 本章小结	212
10.2 面向切面编程	192	10.7 课内实训	212
10.2.1 认识 AOP	192	10.8 实训题	212

第 11 章 基于 Struts 2+Spring+Hibernate 的网上书城项目实训 213

11.1 项目需求说明	213	11.3.1 系统代码介绍	218
11.2 数据库分析	215	11.3.2 系统运行过程介绍	224
11.2.1 E-R 模型图	215	11.4 本章小结	230
11.2.2 数据库表设计	216	11.5 实训题	230
11.3 网上书城设计与实现	218		

教学要求



- 掌握：Struts 2 的安装与配置
- 理解：Struts 2 的开发流程、Action 的实现类以及配置 Action 的方法
- 了解：动态方法调用、类型转换

1.1 配置 Struts 2 运行环境

经过多年的发展，Struts 1 已经成为一个高度成熟的框架，其稳定性和可靠性都得到了广泛的证明。市场占有率超过 20%，拥有众多的拥趸，几乎成为工业标准。但是随着时间的流逝，技术的进步，Struts 1 的局限性也越来越多地暴露出来，并且制约了 Struts 1 的继续发展。

目前，Struts 已经分化成两个框架：第一个是在传统的 Struts 1 的基础上，融合了另一个优秀的 Web 框架 WebWork 的 Struts 2。Struts 2 虽然是在 Struts 1 的基础上发展起来的，但是其实质上是以 WebWork 为核心的。Struts 2 为传统的 Struts 1 注入了 WebWork 的先进的设计理念，统一了 Struts 1 和 WebWork 两个框架。Struts 1 分化出来的另外一个框架是 Shale。这个框架远远超出了 Struts 1 原有的设计思想，与原有的 Struts 1 关联很少，使用了全新的设计思想。Shale 更像一个新的框架而不是 Struts 1 的升级。本书主要学习 Struts 2 框架。

要使用 Struts 2 框架进行 Web 开发或者运行 Struts 2 程序，就必须先配置 Struts 2 的运行环境。

配置 Struts 2 运行环境首先就是配置 JDK 环境变量，然后下载安装 Struts 2 框架。至于 Web 服务器，则选择开源的 Tomcat。上述准备工作完成后，在下一节将使用 Struts 2 进行一个简单实例的开发。

访问 Java 官方网站 <http://java.sun.com>，可以直接下载 JDK。JDK 的安装很简单，这里不再叙述。在安装和配置 JDK 后，配置 JDK 环境变量，然后就可以安装 Java Web 服务器了。Tomcat 服务器的官方网站是 <http://tomcat.apache.org/>。该网站提供了 Tomcat 的直接使用版和安装版的下载链接，目前最新版本是 Tomcat 7.x。

Struts 的官方网站 <http://struts.apache.org/>，提供了 Struts 2 框架的下载链接，目前最新版本是 Struts 2.x。下载时有多个选项可供选择，本书选择 Full Distribution 选项，即 Struts 2 的完整版。

下载完毕后，将下载的压缩包进行解压，然后找到 lib 文件夹。该文件夹内存放的就是 Struts 2

的核心库和第三方插件库，开发 Struts 2 需要的 API 就是这些类库，也就是把需要的相关类库复制到 Web 应用的 WEB-INF/lib 文件夹中。这里把 struts2-core-2.2.1.jar、xwork-2.2.1.jar、freemarker-2.3.1.jar、commons-fileupload-1.2.1.jar 以及 commons-logging-1.0.4.jar 复制到 Web 应用的 WEB-INF/lib 文件夹中，这几个包的作用如下所示。

- Struts 2-core-2.2.1.jar: Struts 2 的核心库。
- xwork-2.2.1.jar: WebWork 的核心库。
- ognl-3.0.jar: OGNL 表达式语言，Struts 2 支持该 EL 表达式。
- freemarker-2.3.16.jar: 表现层框架，定义了 Struts 2 的可视组件主题。
- commons-logging-1.0.4.jar: 日志管理
- commons-fileupload-1.2.1.jar: 文件上传与下载



由于 Struts 2 的版本不同，分发包中的目录结构及类库可能不相同，但变化不大。

也可以使用 MyEclipse 自带工具加载 Struts 2 框架，如图 1-1 所示。

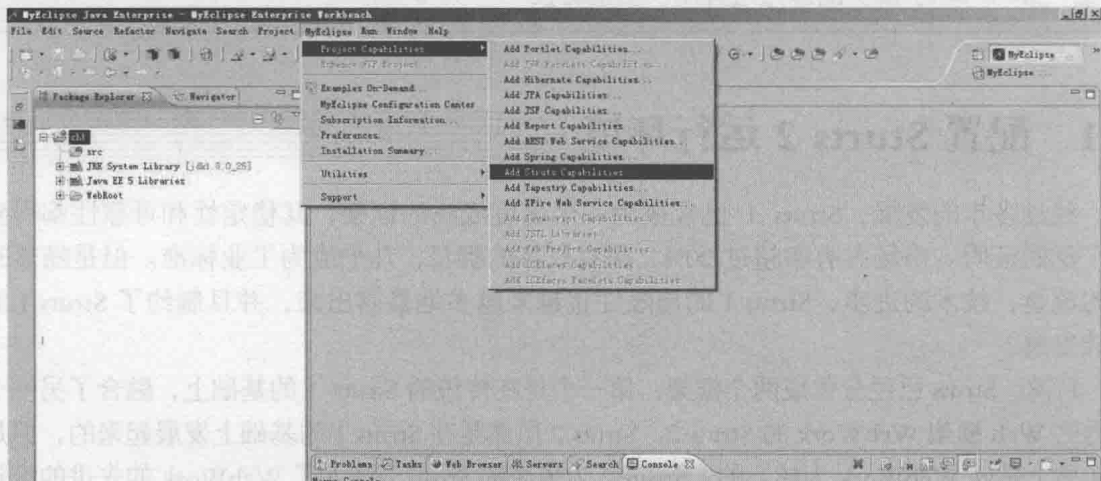


图 1-1 MyEclipse 添加 Struts 框架

添加 Struts 框架支持后，选择 Struts 2 框架，如图 1-2 所示。

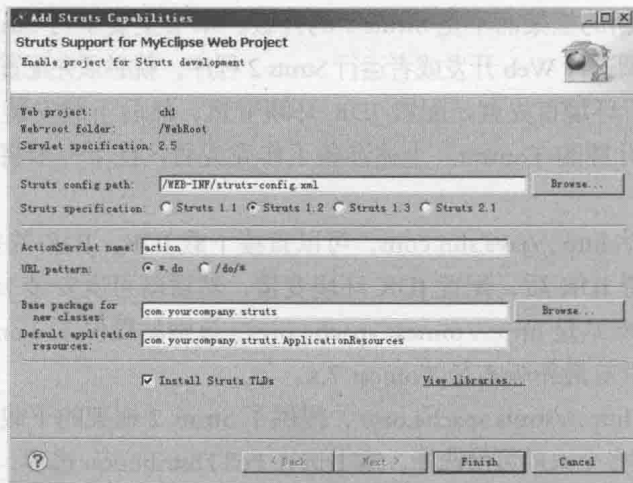


图 1-2 选择 Struts 2 框架

加载完成后，自动添加了 Struts 2 所需 jar 包，如图 1-3 所示。

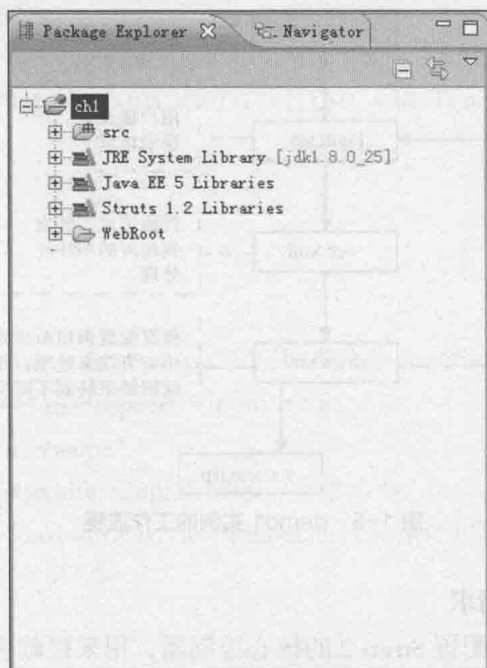


图 1-3 加载完成

1.2 第一个 Struts 2 实例

本实例实现了一个简单的登录功能，即当用户打开一个登录页面后，输入登录名称和登录密码，如果登录名称和登录密码输入正确就进入欢迎页面，否则就重新回到登录页面。图 1-4 所示为本实例的目录结构图，将该项目命名为 demo 1。



图 1-4 demo1 的目录结构



注意

web.xml 文件存储在 WEB-INF 文件夹下，Struts.xml 存储在 classes 文件夹下，lib 文件夹下存储的是所必需的类库包，如 Struts 2 核心类库包等。

demo 1 实例的工作流程如图 1-5 所示。

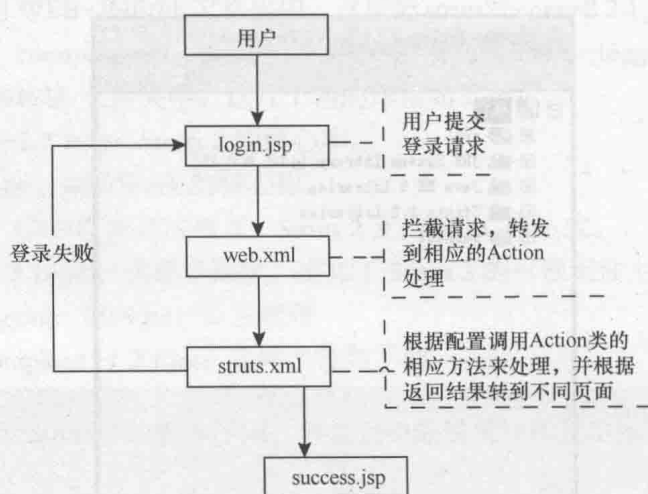


图 1-5 demo1 实例的工作流程

1. Struts 2 拦截用户请求

首先在 web.xml 文件中配置 Struts 2 的核心控制器, 用来拦截客户端请求, 并把请求转发到相应的 Action 类来处理。web.xml 文件的内容如下。

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
  <filter>
    <filter-name>struts2</filter-name>
    <filter-class>
      org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter
    </filter-class>
  </filter>
  <filter-mapping>
    <filter-name>struts2</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
</web-app>
  
```

2. 创建登录页面 login.jsp

Struts 2 的标签库是 Struts 2 的重要组成部分, 它提供了既丰富又强大的功能。这些标签库不仅为表现层提供了数据处理能力, 而且提供了基本的流程控制功能, 以及对国际化、Ajax 的支持等诸多功能。通过使用 Struts 2 的标签库, 可以最大限度地减少登录页面的编码量。login.jsp 使用 Struts 2 标签库实现了一个表单。login.jsp 的内容如下。

```
<%@ page language="java" import="java.util.*"
pageEncoding="UTF-8"%>
<%@ taglib prefix="s" uri="/struts-tags" %> <!-- 导入 Struts2 标签库 -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<style type="text/css">
*{font-size: 12px;}
</style>
<html>
  <body>
    <div style="font-size:14px; font-weight:bold">用户登录</div>
    <div><s:form action="" namespace="/login">
      <s:textfield name="username"
style="font-size:12px;width:120px;" label="登录名称" />
      <s:password name="password" style="font-size:12px;width:120px;" label="登录密码" />
      <s:submit value="登 录"/>
    </s:form>
    </div>
  </body>
</html>
```

3. 创建欢迎页面 success.jsp

success.jsp 为登录成功后显示的欢迎页面。success.jsp 的代码如下。

```
<%@ page language="java" import="java.util.*"
pageEncoding="UTF-8"%>
<html>
  <body>
    <div style="margin:40px;font-weight:bold">
      登录成功，欢迎您！
    </div>
  </body>
</html>
```

4. 创建业务逻辑控制器 LoginAction

上面两个页面文件是本实例的视图组件，现在来创建本实例的业务逻辑控制器组件 Action。LoginAction.java 的代码如下。

```
package com.action;
import com.opensymphony.xwork2.ActionSupport;
public class LoginAction extends ActionSupport {
  private String username;
  private String password;
  public String getUsername() {
```

```

        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String execute() throws Exception {
        if(this.username.equals("admin")&&this.password.equals("12345"))
            return SUCCESS;
        {
        else {
            return LOGIN;
        }
        }
    }
}

```

5. 配置 LoginAction

当 Action 处理完客户端请求后就会返回一个字符串，每个字符串都对应一个视图。

在 struts.xml 文件中配置 Action 时，用 name 属性定义该 Action 的名称，用 class 定义该 Action 的实际实现类，method 表示 checkLogin 该 Action 由实际实现类 LoginAction 的 checkLogin() 方法来处理。

配置 Action 时，为每个 Action 都指定了 result 元素，每个 result 元素都定义了一个逻辑视图，而用 name 定义了 Action 所返回的字符串。

struts.xml 文件的内容如下。

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE struts PUBLIC "-//Apache Software Foundation//DTD Struts Configuration
2.1//EN" "http://struts.apache.org/dtds/struts-2.1.dtd">
<struts>
  <package name="default" extends="struts-default" namespace="/">
    <action name="checkLogin" class="com.action.LoginAction">
      <result name="success">/success.jsp</result>
      <result name="login">/login.jsp</result>
    </action>
  </package>
</struts>

```

struts.xml 文件应放在 WEB-INF\classes 目录下，至此就创建了本实例的所有文件。把本实

例部署到 Tomcat 服务器上后打开 login.jsp 页面，在两个文本框中都输入 admin，如图 1-6 所示。单击“登录”按钮，登录成功后会转向 success.jsp 页面，如图 1-7 所示。



图 1-6 登录页面

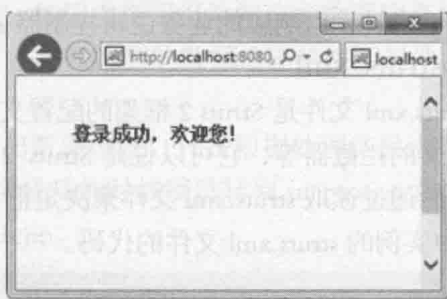


图 1-7 欢迎页面

1.3 Struts 2 框架工作原理

1.2 节讲述了如何实现一个 Struts 2 的登录实例，本节将对这个实例进行剖析，其工作原理如下所示。

(1) 在图 1-6 中单击“登录”按钮，向服务器提交登录请求。

(2) 由于在 web.xml 文件中配置了 Struts 2 核心控制器 FilterDispatcher，因此，此时 Struts 2 会拦截该登录请求。

(3) 由于表单的 action 值为 checkLogin，因此 FilterDispatcher 会在 struts.xml 配置文件中查找 name 为 checkLogin 的 Action 配置，找到后将已经拦截的登录请求转发给 LoginAction 来处理。

(4) 由于在 struts.xml 配置文件中 Action 的 method 属性的值是 Checklogin，因此 LoginAction 会调用自身的 checkLogin() 方法完成对客户端的登录请求的处理。如果登录成功，就返回 SUCCESS，登录失败就返回 LOGIN。



注意

LoginAction 类的 SUCCESS 和 LOGIN 继承自 Struts 2 的 ActionSupport 类，分别对应字符串 success 和 login。

(5) 由于在 struts.xml 配置文件中配置 LoginAction 时指定了 <result name="success">/ success.jsp</result>，因此当 LoginAction 返回 SUCCESS 时转向 success.jsp 页面，否则转向 login.jsp 页面。

以上 5 步概述了 1.2 节实现登录请求的处理流程，如图 1-8 所示。

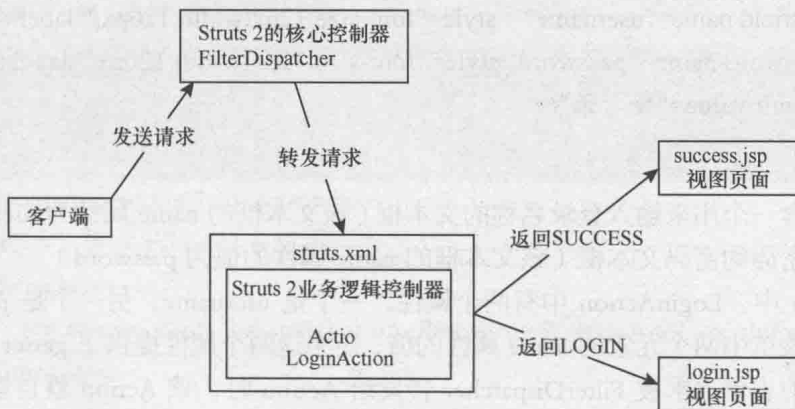


图 1-8 处理流程

1. FilterDispatcher 是 Struts2 的核心控制器，负责拦截所有的客户端请求，作为一个 Filter 通过 web.xml 文件被加入 Web 应用当中，当有客户端请求到达时，它就会进行拦截，然后将客户端请求转发给响应的业务逻辑控制器 Action 进行处理。

2. struts.xml

struts.xml 文件是 Struts 2 框架的配置文件，主要负责配置业务逻辑控制器 Action，以及用户自定义的拦截器等，它可以是 Struts 2 各个组件之间的纽带。FilterDispatcher 在拦截客户端请求后通过读取 struts.xml 文件来决定把该客户端请求转发给哪个 Action 进行处理。下面是 1.2 节中实例的 struts.xml 文件的代码。

```
<include file="struts-default.xml"/>
<package name="default" extends="struts-default" namespace="/">
    <action name="checkLogin" class="com.action.LoginAction">
        <result name="success">/success.jsp</result>
        <result name="login">/login.jsp</result>
    </action>
</package>
</struts>
```

在上述代码中，通过 include 标签引入了 struts-default.xml 这个文件。此文件是 Struts 2 默认的核心配置文件，其中配置了 Struts 2 框架自身所提供的拦截器、类型转换器和数据校验等。而在 Web 应用当中经常要用到这些拦截器、类型转换器、数据校验等组件，所以就先通过 include 标签引入了 struts-default.xml 这个文件。

3. Action

Action 是 Struts 2 的业务逻辑控制器，负责处理客户端请求并将处理结果输出给客户端。要想处理客户端请求，就必须获得请求字符串的参数或从表单提交的数据。

Struts 2 采用了 JavaBean 的风格，即要访问数据，就要给每个属性都各提供一个 getter 方法和一个 setter 方法。获得请求字符串的参数和表单提交的数据也是如此。每一个请求字符串的参数和从表单提交的数据都可以作为 Action 的属性，所以可以通过提供 setter 方法来获得请求字符串的参数或从表单提交的数据。现在把 1.2 节实例中的代码复制过来进行分析，登录页面 login.jsp 的表单代码如下。

```
<s:form action="" namespace="/login">
    <s:textfield name="username" style="font-size:12px;width:120px;" label="登录名称" />
    <s:password name="password" style="font-size:12px;width:120px;" label="登录密码" />
    <s:submit value="登录" />
</s:form>
```

表单中包含一个用来输入登录名称的文本框（该文本框的 name 属性为 username）和一个用来输入登录密码的密码文本框（该文本框的 name 属性的值为 password）。

在代码 1.3 中，LoginAction 中有两个属性，一个是 username，另一个是 password，分别对应登录页面表单中两个元素的 name 属性的值，还为这两个属性提供了 getter 和 setter 方法。当客户端发送的表单请求被 FilterDispatcher 转发给 Action 时，该 Action 就自动通过 setter 方法获得表单提交的数据信息。

4. 视图组件

Action 在处理完客户端请求后会通过视图组件把处理结果显示出来，包括如下两种情况。

- 一种情况是 Action 向视图组件输出数据信息，然后由视图组件把这些数据信息显示出来，例如 Action 执行了查询数据库的操作，并把查询到的数据输出给视图组件，然后由视图组件把这些数据信息显示出来。
- 另一种情况是 Action 并没有向视图组件输出数据信息，只是根据处理结果进行简单的页面跳转。例如在 1.2 节的实例中，当登录成功的时候就跳转到 success.jsp 页面，当登录失败的时候就重新跳转到登录页面 login.jsp。

这些都是在 struts.xml 文件中进行配置的，如下所示。

```
<result name="success"/>/success.jsp</result>
<result name="login"/>/login.jsp</result>
```

在上述代码中可以看到在 Action 中配置了两个 result，即两个视图。一个是登录成功后的视图，另一个是登录失败后的视图。这里 result 的 name 属性的值与 Action 的返回值相对应。



Struts 2 除了支持传统的 JSP 视图外，还提供了对 FreeMarker、Velocity 等视图的支持。

1.4 配置 Struts 2

任何 MVC 框架都需要与 Web 应用整合，这就不得不借助在 /WEB-INF 路径下的 web.xml 文件，只有配置在 web.xml 文件中的 Servlet 才会被应用加载。

通常，所有 MVC 框架都需要 Web 加载一个核心控制器，对于 Struts 2 框架而言，需要加载 FilterDispatcher，只要 Web 应用负责加载 FilterDispatcher，FilterDispatcher 就会加载应用的 Struts 2 框架。

1.4.1 配置 web.xml

因为 Struts 2 将核心控制器设计成一个过滤器 Filter，而不是一个普通的 Servlet，所以为了让 Web 应用加载 FilterDispatcher，只需要在 web.xml 文件中配置 FilterDispatcher 就可以了。配置 FilterDispatcher 的代码片段如下。

```
<!-- struts 过滤器 -->
<filter>
<!--配置 Struts 2 核心 Filter 的名字 -->
    <filter-name>struts 2</filter-name>
<!--配置 Struts 2 核心 Filter 的实现类 -->
    <filter-class>
        org.apache.struts 2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter
    </filter-class>
</filter>
```