

# 基于 STRUTS 2 框架的产品信息发布系统的设计与实现

董袁泉

(沙洲职业工学院, 江苏 张家港 215600)

**摘要:** 在对 STRUTS 2 框架和 MVC 模型分析比较的基础上, 阐述了 STRUTS 2 框架的实施方法, 将 STRUTS 2 框架应用于实际开发, 建立了产品信息发布系统模型, 并分析了开发中的关键问题。

**关键词:** STRUTS 2; MVC; 信息发布系统

中图分类号: TP311 文献标识码: A

## Design and implementation of product information publishing system based on STRUTS 2

DONG Yuan Quan

(Shazhou Polytechnical Institute of Technology, Zhangjiagang 215600, China)

**Abstract:** This article analyzes and compares STRUTS 2 framework and the MVC model, and explains the implementation of STRUTS 2 framework; then it applies STRUTS 2 framework to the actual development, establishes a product information system model and analyzes the development of the key issues.

**Key words:** STRUTS 2; MVC; product information publishing system

随着互联网技术的飞速发展, JavaEE 以其开放性、扩展性以及各大厂商的支持, 逐渐成为 Web 应用软件的开发标准。近几年来, STRUTS 框架作为 MVC 模型的一个良好实现, 使基于 JavaEE 的 Web 应用的开发能够获得更好的可扩展性、可重用性、易开发性和易维护性。STRUTS 出现至今, 自身也在不断地成熟和完善, STRUTS 2 (WebWork 2) 的发布为 JavaEE Web 应用的开发者提供了更好的 MVC 实现和更加简单易用的实施方法; 采用 STRUTS 2 框架, 以及 JSP、Servlet、JavaBean、JDBC 等技术开发本产品信息 Web 发布系统, 能够充分发挥 STRUTS 2 框架的优势, 使得系统层次清晰, 保证了系统的稳定高效, 并获得了较好的扩展性和易维护性。

### 1 STRUTS 框架概述

STRUTS 是 Apache Jakarta 项目组的一个 Open-Source 项目, 是 MVC 模型的一个良好实现。MVC (Model-View-Controller) 模型把一个应用的输入、处理、输出流程按照 Model、View、Controller 的方式进行分离, 使得应用程序设计过程更加清晰, 提高了系统的灵活性和扩展性。MVC 应用于 Web 发展出了 MVC 2 模型, 在 MVC 2 模型中, View 由 JSP 来担当, Controller 由 Servlet 来担当,

Model 由 JavaBean/EJB 来担当。Servlet 先接受客户端的请求, 然后调用相应的 Bean 进行业务逻辑处理, 根据处理结果再调用相应的 JSP 页面, JSP 页面根据存储结果的 Bean 生成结果页面, 返回给客户端。STRUTS 框架包括一组相互协作的类 (组件)、ActionServlet 以及 JSP Taglib 标签库, 实现了 MVC 2 模型中的 Controller 和 View; 而 MVC 2 模型中的 Model 必须由开发者自己来实现, STRUTS 提供了抽象类 Action, 使开发者能够将 Model 应用于 STRUTS 框架。

### 2 STRUTS 2 体系结构

STRUTS 2 是 STRUTS 1 的更新版本, 对 STRUTS 1 进行了改进和升级, 例如用 Action 接口取代了 Action 抽象类, 为每一个请求创建一个 Action 对象, 去掉了 ActionForm 类, 用重新定义的 struts.xml 替代了 struts-config.xml, 数据校验方法更加灵活以及采用了分层的国际化方法等。STRUTS 2 更好地实现了 MVC2 模型。其体系结构图和处理流程如图 1 所示。

在 HttpServletRequest 到达 Servlet 容器之前要经过一系列过滤器链, 包括 ActionContextCleanUp 过滤器等; 然后调用 FileterDispatcher, FileterDispatcher 参考 ActionMap-

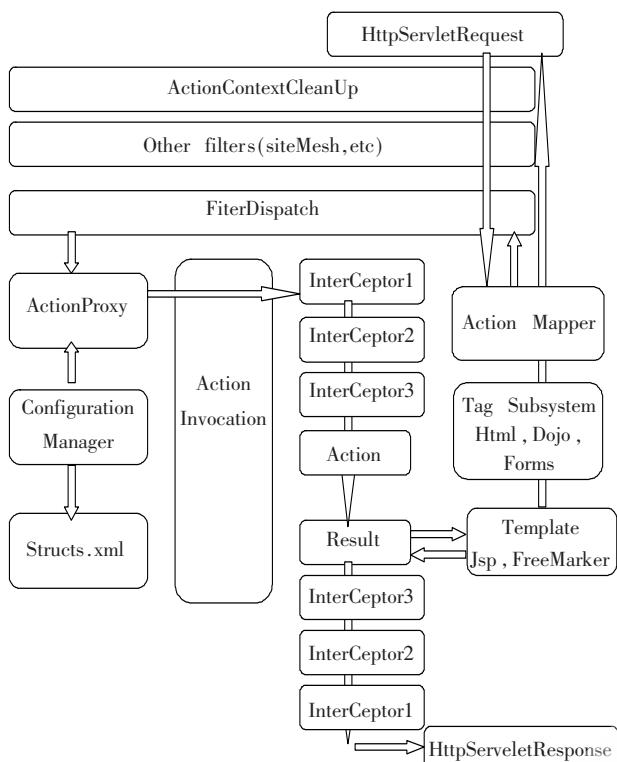


图 1 STRUTS2 体系结构

per 来决定请求是否与相应 Action 相关联。

如果 ActionMapper 确定了请求对一个 Action 的调用, 它就会将控制权委托给 ActionProxy, ActionProxy 创建对应的 Action 实例, 然后 ActionProxy 参考由 struts.xml 配置的 ConfigurationManager 生成 ActionInvocation。之后, Action 和 Interceptor 的执行都由 ActionInvocation 来控制。在 Action 执行前可能会执行多个 Interceptor。

当 Action 执行结束后, ActionInvocation 根据 struts.xml 中的 result 配置查找相应的结果页面, 并将 Action 的执行结果输出到结果页面中。这个页面一般是使用 STRUTS 提供的标签库编写的 JSP 页面。

接下来, ActionInvocation 控制 Interceptor 以反序执行。最后, HttpServletResponse 根据 Web.xml 中的配置通过相应的过滤器, 返回客户端。

### 3 STRUTS 2 框架的实施

#### 3.1 Action 类的实施

在 STRUTS 1 中, 必须继承 Action 抽象类, 才能将应用集成到 STRUTS 框架, 而在 STRUTS 2 中的实施方法相对简单, 只需要实现 Action 或 ActionSupport 接口。而且 STRUTS 2 取消了 ActionForm 类, 直接在 Action 类中保存页面提交的输入信息和需要提交给页面的输出信息, 对于数据的操作更加简便。在 execute()方法中可以直接处理这些信息, 并实现对数据库的查询、增加、删除和修改。

#### 3.2 映射关系的配置

STRUTS 2 使用重新定义的关键配置文件 struts.xml

来代替 STRUTS 1 中的 struts-config.xml, 通过 struts.xml 控制 Action 请求到 Action 类的映射, 以及 Action 类到 JSP 页面的映射关系。在 struts.xml 文件中, 主要使用三种标签定义这些映射关系:

(1) 使用<package name=" " namespace=" " extends=" " > 标签, 配置 Web 应用中用到的 Action 类的包的相关属性。name 属性指定 Action 类的包名。

(2) 使用<action name=" " class=""> 标签, 配置 Action 类的相关属性。其中 name 属性指定 Action 的类名, class 属性用“包名.类名”的格式指定 Action 类完整的包名和类名。

(3) 使用<result>标签配置某个 Action 类的输出页面。在<result>标签的内容中, 以“/”开头表示以发布目录作为当前目录。

#### 3.3 标签库和国际化的实施

在 STRUTS 框架中, JSP Taglib 标签库和国际化 (Internationalization) 是对 MVC 模型中 View 模块的实现。通过在 JSP 页面中使用框架提供的标签库, 可以实现显示 Action 中动态数据的目的。对于国际化, STRUTS 1 将所有的资源簇 (Resource Bundle) 保存在一个本地资源文件 application.properties 文件中, 而 STRUTS 2 提供了分层的本地资源文件, 我们可以将用到的资源簇分别存放在这些资源文件中, 框架可以通过标签属性和 getText() 等方法使用资源文件的内容。这些资源文件的搜索顺序是: ActionClass.properties → BaseClass.properties → Interface.properties → package.properties。在开发过程中通常使用 package.properties 保存一个 package 中用到的资源簇。在这些资源文件中, 其内容以“关键字=值”的形式保存, 如“username=User Name”。

### 4 系统的设计实现

#### 4.1 系统的层次结构

产品信息发布系统总体上采用“浏览器/Web 服务器/数据库”的三层体系结构。系统的层次结构如图 2 所示。

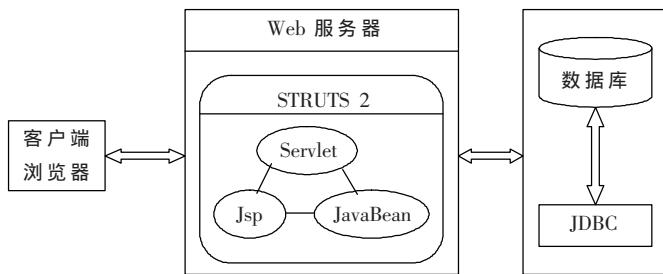


图 2 系统层次结构图

Web 服务器接受请求, STRUTS 框架负责请求的处理, 在框架内通过特定的映射关系调用相应的业务逻辑, 并将结果以 JSP 页面的形式返回客户端浏览器, 实现与客户端的交互。数据库服务器用来存储系统用到的所有数据, 数据可以由管理员通过数据库管理程序录

入,也可以由拥有权限的客户通过客户端进行管理。系统的客户端只需要一个浏览器,不同的用户通过浏览器对数据进行查询、增加、删除、修改操作,对信息进行管理。

#### 4.2 系统功能结构设计

根据需求分析,用户可分为一般用户、注册用户、商家用户和系统管理员。一般用户进入该网站后,可以浏览该网站的商品内容,查找需要的商品信息和商家信息;注册用户登录后,除了拥有一般用户的全部功能以外,还具有登记缺货、向商家留言、联系商家等功能;商家用户登录后,除了拥有注册用户的全部功能外,还具有查询缺货、发布新产品、产品下架等功能;系统管理员具有系统的全部功能的使用权限,可以查看所有用户资料,核准商家注册,删除用户和商家信息等。

#### 4.3 数据库设计

系统使用 SQL Server2000 作为后台数据库,对数据信息规范化后可以得到以下几个数据库表:

user:(用户 ID, 用户名, 真实姓名, 密码, 性别, 地址, 电话, e-mail, 地址, 密码提示问题, 密码问题答案, 简介, 权限)用来记录用户信息,其中“权限”表明该用户的使用权限是注册用户、商家或者是系统管理员。

mobilePhone:(手机编号, 品牌, 型号, 产地, 配件, 技术指标, 上市时间 ...)用来记录各商家所有手机产品的信息。

B2PMap(Boss2ProductMap):(商家 ID, 手机编号, 上架时间, 下架时间, 库存, 售价)用来记录单个商家发布的手机产品的信息。

OOS(OutOfStock):(编号, 手机名称, 品牌, 型号, 登记时间, 用户 ID)用来记录用户登记的缺货信息。

#### 4.4 系统主要功能模块的实现

产品信息发布系统为众多手机销售商家提供了一个统一的信息发布平台,商家用户登录后可以将自家销售的产品信息发布到本系统上供其他用户查看,注册用户可以通过该系统登记缺少的产品,商家可以根据登记的缺货调整自家的上架产品。因此,商家是本系统的主要用户对象,产品信息发布功能是本系统的核心功能。

(1)商家发布产品信息的程序文件是 Phone\_info\_publish. class, 该类实现了 ActionSupport 接口, 用相应的属性保存了 JSP 页面提交的产品信息数据, 实现了相应属性的 getter 和 setter 方法, 并在 execute() 方法中处理这些数据, 将产品信息写入数据库, 主要代码如下:

```
Context initCtx=new InitialContext();
Context ctx=(Context) initCtx.lookup("java:comp/env");
Object obj=(Object) ctx.lookup("jdbc/PlatformDB ");
this.ds=(DataSource) obj;
this.conn=this.ds.getConnection();
this.stmt=this.conn.createStatement()
```

ResultSet.TYPE\_SCROLL\_SENSITIVE, ResultSet.CONCUR\_UPDATABLE);

this.stmt.executeUpdate(sql);

(2)struts.xml 中映射关系配置,部分代码如下:

```
<package name = " platform" namespace = "/platform"
extends="struts-default">
<action name = "Product_info_public" class =
"platform. Product_info_public">
<result>/platform/productList.jsp</result>
</action>
```

(3)productList.jsp 页面使用 Taglib 显示动态数据,部分代码如下:

<table border="1">

```
<tr><th>手机编号</th><th>品牌</th><th>型号</th><th>产地</th><th>配件</th></tr>
```

<logic:present name="result">

```
<logic:iterate id="phone" name="result" type =
"platform. Phone">
```

<logic:present name="phone">

```
<tr>
<td><bean:write name="phone" property =
"phoneID"/></td>
<td><bean:write name = " phone " property =
"brand"/></td>
```

```
<td><bean:write name = " phone " property =
"phoneType"/></td>
```

```
<td><bean:write name = " phone " property =
"producingArea"/></td>
```

```
<td><bean:write name = " phone " property =
"accessory"/></td>
```

</tr>

产品信息发布系统采用 STRUTS 2 以及 JSP、Servlet、JavaBean 和 JDBC 等技术,实现了 MVC 模型,使得整个系统开发过程层次清晰,模块划分明确,增强了系统的稳定性。由于实施了 STRUTS 2 框架,各个组件可以很容易地升级和更换,不会对整个系统造成影响,使整个系统具有了更好的可扩展性和易维护性。

#### 参考文献

- [1] <http://struts.apache.org/2.x/index.html>.
- [2] SHALLOWAY A, Trott J R. Design Patterns Explained [M]. 北京: 清华大学出版社, 2004.
- [3] 孙卫琴.精通 Struts: 基于 MVC 的 JavaWeb 设计与开发 [M]. 北京: 电子工业出版社, 2004.
- [4] [http://www.diybl.com/course/1\\_web/webjs/2008224/101769.html](http://www.diybl.com/course/1_web/webjs/2008224/101769.html).

(收稿日期: 2009-02-11)