

学生应知信息知识

JAVA 知识手册 (十)

狄登峰 主编

# 目 录

用 Struts 开发基于 MVC 的 Web 应用 .....	1
Spring Framework 标记库初学指南 .....	32
Java 中文档对象模型 DOM 经验小结 .....	44
JBuilder 设计获取图片属性程序 .....	68
JXTA 技术网络协议概述 .....	93
WEB 应用程序的测试与优化概述 .....	96
JXTA 技术的设计目标和层次结构 .....	101
Java 语言中两种异常的差别 .....	107
JXTA 技术与应用发展概述 .....	114
Java 用户界面本地化实例探索 .....	116
JFreeChart 在 Webwork 中的应用 .....	130
用 Java Web 服务打造聊天程序 .....	146

## 用 Struts 开发基于 MVC 的 Web 应用

### MVC 介绍

MVC 模式是一种非常理想化的设计模式，应用 MVC 模式完成两个以上项目的人都有同样的体会，他们已经对以前的工作方法进行了彻底的改造。工作模式的改变要付出痛苦的代价，但现在你有现成的技术架构可以采用，避免在项目中自己开发、摸索。它就是开源 Apache Struts framework，它提供了实现 MVC 设计模式最好的实现工具。

在本文中，我们将简单了解、体会一下模型-视图-控制器（MVC）设计模式，特别地，我们来看看如何用 Struts 架构来完美地实现 MVC 模式。我们先从理论上简单地描述 MVC 模式，然后用我们一个简单的例子来实现我们自己的 MVC 架构。在对 MVC 模式有了了解后，我们用 Struts 来看这个新技术是如何帮助我们迅速、简单地创建基于 MVC 的 Web 应用。

### 模型-视图-控制器(Model-View-Controller)模式

MVC 模式最早是在 Smalltalk(一种面向对象的语言)这种程序语言设计中被提出来的。我们暂时先忽略它的历史，集中注意力在关注它怎样被应用在 Web 应用开发中。

当 Java 的 Servlets 技术最开始出现的时候，程序员们立刻意识到这是一项极其有用的技术。与同时代的 CGI Web 开发技术相比，Servlets 更快，更灵活，更可靠，更强大。然而，开发基于 Servlets 技术的 Web 应用有一个巨大的缺陷--需要使用例如 `out.println` 之类的语句来输出浏览器识别的 HTML。频繁使用这个方法是个错误的倾向，开发极其浪费时间（程序员需要经常退出所有应用程序进行重新编译）。并且这也使修改 Web 页面的工作也变得很困难，因为 Web 的表现和逻辑在一堆令人恐惧的代码中掺乎在一起。

于是作为解决方法的 JavaServer Pages (JSP) 出现了，它们将 Servlets 变成它们运行的结果。应用 JSP 技术，我们将业务逻辑用一系列夹杂在 HTML 中的 `<%>` 标识来表达。以开发 JSP 为核心的应用尽管比以 Servlet 为核心的应用有进步，但看起来仍然是杂乱无章的，仍然需要用额外的代码来控制应用页面的流转。在充满格式化代码的 JSP 页面上，没有地方来增加这样额外的控制代码。显然需要寻找别的出路。

不久人们认识到同时应用 JSP 和 Servlets 两种技术开发 Web 应用是一种不错的选择。毕竟，Servlets 擅长处理业务逻辑的编程，处理请求，控制功能页面的流转，而 JSP 则是格式化请求处理结果，通过浏览器获得用户输入。这种工作机制后来变成了人们长说的 Model2（用 JSP 或 Servlets 中单独的一种实现 web 应用被称做 Model 1）。

Model 2 不是一项革命性的新模式，其实它是来自

于 Smalltalk 语言研发过程中出现的 MVC 模式。大多数情况下，Java 程序员趋向于可完全互换地使用这两个名词。

什么是 MVC 模式？

此前我们已对 MVC 在开发基于 Java 技术 Web 应用中的使用历史有了初步的了解，现在让我们来看看这种模式的细节。本节中，我们来准确地了解一下 Models、Views、Controllers 的确切含义，它们实现的任务，以及如何利用它们实现一个简单的 MVC 框架。我们先来看看 Model、View、Controller 是如何交互工作的。

图 SM01

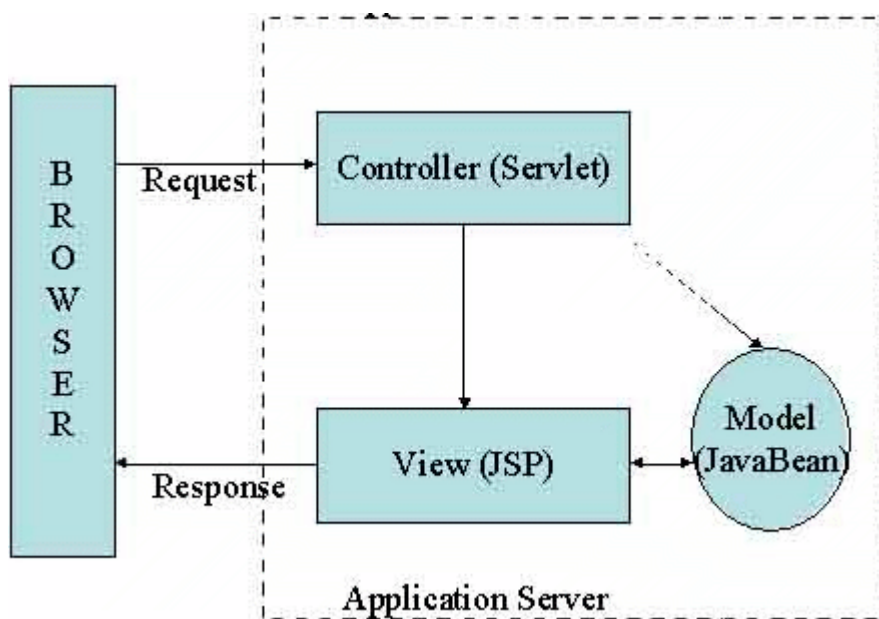


Figure 1 : Model 2/MVC 架构

如上图所示，用户通过提交 requests 与 Controller

组件 (通常表现为 Servlets ) 交互。接着 Controller 组件实例化 Model 组件 (通常表现为 JavaBeans 或者类似技术), 并且根据应用的逻辑操纵它们。一旦 Model 被创建, Controller 决定下一个为用户显示的 View (常常表现为 JSP), 同时 View 与 Model 交互操作, 获得并为用户显示相关数据。在它被提交到 Controller 重新开始此操作之前, View 可以修改 Model 的状态。

为了更全面得理解组件之间的交互, 我们来看一个应用这种框架实现的简单例子。这是一个完成提交、记录用户登陆信息的简单应用。

### View

本例的 View 由两个简单的 JSP 页面组成。请参考代码 (login.jsp、welcome.jsp)。

1 > login.jsp 只是简单地提供了用户输入姓名和口令的操作界面。输入完成后, 登陆页提交输入到 controller Servlet (代码如后 Controller 部分说明), 告诉它需要调用"登陆操作 (login action)" (操作参数通过 form 来传递);

2 > welcome.jsp 页面利用用户前页提供的用户姓名显示一个欢迎信息。这里只是简单地调用了 session 中的 JavaBean (从 userBean 的 tag 标识可以看到)。这个 Bean 是被 Controller 置于 session 中, 我们接下来可以看到。

### Controller

样例中的 controller 由一个 Servlet 构成，代码参见 ( Controller.class )。实现了我们应用中的 Controller。

这是个简单的 controller，仅仅根据一个 request 参数( action )决定调用哪一个 action。本例中，页面将 login action 作为参数传递进来，所以 LoginAction 被调用。该 action 实现了一个标准接口 ( Action )，定义了将 Request 和 Response 对象作为参数的 execute 方法。这个 action 类返回被调用的下一页的路径，于是用户重定向到此页面。

LoginAction 类从 request 中获得 username 参数，创建一个新的 model 对象( UserBean )，并将其传至 Session，并返回 "/welcome.jsp" 标识流转的下一页面是 welcome.jsp。

Model.

我们示例中的 model 也很简单，仅由一个 JavaBean 构成。代码参考 UserBean.class。

Action 的扩展应用

如你所示，这是一个很简单的 Model 2 应用，但它可以被在更大程度扩展。比如，我们可以动态配置映射 request 参数的 action，我们也可以具体化 controller 的流转控制（比如 action 可以通过一个配置管理器 ( configuration manager ) 来动态获得需要返回的页面，而不是象现在这样写死在程序里）。

然而,事实上有一个现成的框架提供所有这些控制、MVC 组装相关的可配置项,甚至更多。

这个现成的框架就是 Struts。

### Struts 介绍

Struts 项目作为一个设想是 Craig McClanahan 2000 年提出的,目标是为利用 Java 技术开发基于 MVC 模式的 Web 应用提供一个标准模式。Struts 1.0 在 2001 年中期被最终发布,现在成为 Apache Foundation 的 Jakarta 项目的一部分。Struts 应用范围极广,可以用在不同的项目,不同的行业(我所见到的从电信到电子商务都有应用实例)。

Struts 是一个高度可配置、高度扩展性的 MVC 框架,我们几乎可以用它开发任何能想到的用 Java 技术的 Web 应用。MVC 模式的每一部分在 Struts 中都有相关对应部分。

### Struts 的安装

可 以 在 <http://apache.get-software.com/jakarta/struts/binaries/jakarta-struts-1.1.zip> 下载获得 Struts 的最新版本(目前是 1.1)。下载后解压 zip 文件。发布包中包含了所有开发 Struts 应用所需的类库。发布包的 Webapps 目录下有一个空白的 Struts Web 应用(struts-blank.war),它已经包含了一个 Web 应用的骨架,非常有用,在这个基础上建立自己的应用显然对初学者能很快得到成就感。

自己的代码放在 WEB-INF/classes 目录下，根据自己的需要修改配置文件 WEB-INF/struts-config.xml,做到这一步，Struts 的配置就完成了。现在就拥有了一个完全有效的 Struts 应用了。

让我们来看看 Struts 提供的组件

### View 层

大多数 Struts 应用的 view 层是由 JSP 组成的。为了使 view 的开发更加容易，Struts 提供了一整套 JSP 自定义的 tag 库。这些 tag 库使我们能很容易地提供完全国际化的用户界面，这些界面通常是与 Struts 应用中的 Model 组件交互。

通常 Web 应用的动态前端都是基于 HTML 表单的，这些应用的用户需要应用的可靠性得到保证，这样就需要表单校验。如果用标准的 JSP，记录表单的内容和从一个 JavaBean 获得表单内容简单乏味而且容易出错。Struts 应用 FormBean 使表单处理和校验变得容易。FormBean 与 Struts 的 tag 库结合，使带 form 的 View 开发变得容易而自然。

下面是一个 Struts 的 JSP 页面样例。

```
< %@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" % >

< html:html >
```

```
< head > < /head >
```

```
< body bgcolor="white" >
```

```
< html:errors/ >
```

```
< html:form action="/logon" >
```

```
< table border="0" width="100%" >
```

```
< tr >
```

```
< td >
```

Username:

```
< /td >
```

```
< td >
```

```
< html:text property="username"/ >
```

```
< /td >
```

```
< /tr >
```

```
< tr >
```

```
< td >
```

Password:

< /td >

< td >

< html:password property="password"/ >

< /td >

< /tr >

< tr >

< td >

< html:submit/ >

< /td >

< td >

&nbsp;

< /td >

< /tr >

< /table >

< /html:form >

< /body >

< /html:html >

从以上 JSP 样例可以看出，这与标准的 HTML 表单不同。页面中没有杂乱无章的 JSP `<%>` 代码，然而它却能完成更多的功能。此 JSP 引入了 Struts 的 HTML tag 库，它增加了能够完成收集提供了校验、错误处理、model 交互功能的表单。注意 `<html:errors>` tag，它可以显示 model 或者 controller 已经注册的错误。`<html:form>` tag 则创建了一个基于 ActionForm 对象的 HTML 表单。上例中表单的 action 被置于 `/login`，我们用这个值到配置文件（示例如后）中去找对应的 ActionForm。这种映射关系由表单对象的名字和它被存储的范围（session, 页面，应用等等）组成。对象的属性用 `<html:text>`、`<html:password>` tag 来表示，构成表单。值得夸耀的好处是，ActionForm 被提交时可以自动拾获对应的表单数据，无须我们操心。

从前面我们提到的 ActionForm 的用处来看，它似乎应该被视为应用的 Model，然而事实上他们应当被当作应用中的 controller 的一部分。ActionForm bean 中显示了 model 的属性，但它们不包含任何持续性逻辑或者业务逻辑。ActionForm 只是用来在 Model、View 之间传递 Model 信息。

因为 ActionForm 属于 controller 中的一部分，我们将在后面 controller 的部分来具体了解它。

## Model 层

Struts 应用中的 model 层可以应用任何基于 Java 的技术实现,比如 EJB, Hibernate, 或者 JDO。通常, model 是作为包含数据和业务逻辑的简单 JavaBean 出现的。如前所述 ActionForm 对象不是真正 model 层的体现,同时 Model 层应当独立于 HTML 的表单对象。如果可能的话, model 对象的开发应当是与使用的开发技术和开发环境(Struts 或者其他)无关的,这样我们就可以在不同的环境和应用中很容易地重用它们。

为了演示,我们开发了一个简单的基于 JavaBean 的 model 层对象,它不包含持续性逻辑。这个对象与我们的 ActionForm 对象映射,未来使用 model 层对象时,我们只需用更复杂的逻辑来代替它。

### Controller 层

Struts 内置一个实现了 controller 主要功能的 Servlet,它提供将需要调用的 URL 与一个 action 对象对应起来的功能。这个 Servlet 被称作 ActionServlet,完成下列功能:

- 1 > 根据用户要求决定需要的 action;
- 2 > 为 View 提供 View 需要的数据;
- 3 > 决定要显示的下一个 View。

ActionServlet(强调:该 Servlet 已由 Struts 实现,是 Struts 架构的核心所在,开发者无须关心)的重头工作是调用一系列简单的 Action 类。Struts 开发人员的工作主

要是提供这些 actions 来实现应用的逻辑。创建 action 必须实现 action 接口。此接口包含以下方法：

```
public ActionForward execute(ActionMapping  
mapping,  
  
ActionForm form,HttpServletRequest  
request,HttpServletResponse response)  
  
throws Exception;
```

如上所示，该方法将 ActionForm 作为它的一个参数。上面提到的 ActionServlet 保证了正确的 form 传递给这个方法。在 View 层我们说过，ActionForms 在 Model 层和 View 之间传递数据。

ActionForms 是一个非常简单的对象；以下代码显示了我们将在一个简单的 HTML 表单中用到的 ActionForms：

```
import javax.servlet.http.HttpServletRequest;  
  
import javax.servlet.http.HttpServletResponse;  
  
import org.apache.struts.action.ActionForm;  
  
import org.apache.struts.action.ActionMapping;
```

```
public class Login extends ActionForm {  
  
    protected String username;  
  
    protected String password;  
  
    public void setUsername(String username) {  
  
this.username = username;  
  
    }  
  
    public String getUsername() {  
  
return username;  
  
    }  
  
    public void setPassword(String password) {  
  
this.password=password;  
  
    }  
  
    public String getPassword() {  
  
return password;  
  
    }  
  
}
```

Action 还包含一个 ActionMapping 对象。它被 ActionServlet 自动处理，体现应用的配置。具体配置可以从一个 XML 文件获得，通常就是 struts-config.xml 文件，下面将会提到。

action 运行方法将标准的 request、response 作为参数，应用可以利用这些调用参数。action 类处理完毕后，将 ActionMapping 所映射的要调用的下一个页面作为参数返回给 ControllerServlet ( Struts 内置 )。

### 集成 Struts 组件

我们来看 Struts 是如何将这三层的组件组合在一起构成完整的应用。Struts 应用用 struts-config.xml 来完成配置。这个配置文件包含了应用的所有可配置信息，包括：

1 > 要用到的 controller

2 > ActionForms 和他们对应的 HTML forms

3 > Actions

4 > ActionMappings，它控制应用的整个功能流转

struts-config.xml 的重要配置元素都包含在 < struts-config > 标识下。

```
< ?xml version="1.0" encoding="ISO-8859-1" ? >
```

```
<!DOCTYPE struts-config PUBLIC
    "-//Apache Software Foundation//DTD Struts
    Configuration 1.2//EN"
    "http://jakarta.apache.org/struts/dtds/struts-config_1_2.dtd"
>

< struts-config >
```

配置 ActionForm 对象：

```
< form-beans >

< form-bean

name="logonForm"

type="com.samjdalton.struts.LogonForm"/>

</form-beans >
```

以上配置声明一个"logonForm"的表单，接着说明需要用 com.samjdalton.struts.LogonForm class 来完成该表单设置。

下一步，我们声明 ActionMappings。