

基于JSF实现MVC模式的Web应用

宋丽^{1,2}

(1. 东北林业大学, 黑龙江 哈尔滨 150040; 2. 牡丹江师范学院, 黑龙江 牡丹江 157012)

摘要: JSF是一种新的开发Web应用程序用户界面的框架,它旨在提高Web应用程序的开发效率. 本文介绍MVC模式和基于MVC模式的JSF架构,并举例介绍基于JSF框架实现MVC模式的Web应用,体现了JSF架构在构建可维护性和可扩展性等方面很强的技术特点.

关键词: MVC模式; JSF框架; Web应用

[中图分类号] TP4

[文献标识码] A

[文章编号] 1003-6180(2009)01-0015-02

JSF(Java Server Faces), 或简称Faces, 通过将丰富的、强大的UI(User Interface)组件(如文本框、列表框、分页面板和数据网格等)的支持引入Web开发领域,使得Web开发更加容易. 目前JSF是Java EE5.0的实现标准. 由于JSF目前还算比较新的框架技术,组件还在不断扩展和完善中,JSF抽象了Web开发中的许多细节,在很多方面领先于Struts并可与Struts集成. 相信在不久的将来,JSF将成为Java Web开发中的下一主流.

1 MVC模式

SUN公司先后制订了两种规范,称为JSP Model1和JSP Model2,它们是对采用JSP技术构成Web应用不同模型描述.

1.1 Model1

在使用Java技术建立Web应用的实例中,Web应用就是由一组JSP页面构成的,在JSP页面中可同时实现显示、业务逻辑和流程控制,从而可以快速地完成应用开发. Model1是以JSP为中心的开发模型,这种开发模式在进行快速和小规模的应用开发时有非常大的优势,但是从工程化的角度考虑,它不利于应用扩展和更新,不利于应用系统业务的重用和改动.

1.2 Model2

Model2表示的是基于MVC模式的框架. MVC是Model-View-Controller的简写. “Model”代表的是应用的业务逻辑(通过Java Bean, EJB组件实现),“View”是应用的表示面(由JSP页面产生),“Controller”是提供应用的处理过程控制(一般是一个Servlet). 通过这种设计模型把应用逻辑、处理过程和显示逻辑分成不同的组件实现. 这些组件可以进行交互和重用,从而弥补了

Model1的不足.

MVC模型—视图—控制器模式的核心思想,是将整个程序代码分成相对独立而又能协同工作的三个组成部分. Model层主要实现系统的业务逻辑与应用数据,它是应用程序的核心. View层把数据或行为显示给应用的用户,这是用户与应用程序直接交互的部分. Controller层是Model层与View层之间沟通的桥梁,它可以响应用户的请求并选择恰当的View用于显示,同时它也可以解释用户的输入并将它们映射为Model层可执行的操作. 这种模式的思想在Java应用程序得到了广泛的应用.

2 JSF框架

JSF技术是一个用于构建Web应用程序的服务器端组件框架,它提供了一种以组件为中心来开发Web应用的服务器端技术. 基于JSF构架的Web应用程序符合MVC的设计标准,可以说是MVC设计模式的一种实现. JSF的优势主要在于它是基于MVC设计模式,把视图与模型相分离,JSF网页设计者只需要专注于页面的设计,应用程序开发者主要负责模型部分的JavaBean和EJB的开发,而程序的流程控制则由faces-config.xml配置,使得JSF程序更加易于管理. JSF遵循业务逻辑和显示的分离,JSF架构遵循MVC设计模式的标准. 同Struts等其他架构相比,JSF架构的出现不是要取代这些架构,而是着力于解决现有架构不能解决的难题,如View和Model无法实现完全脱离、客户端类型较多和对UI组件支持不足等问题.

基于MVC模式的JSF架构的结构图参见文献[1],按照MVC模式可以分为以下三个部分:

Model层 JSF的Model层主要用JavaBean或EJB实现,也可以是其他的业务处理对象. 如

收稿日期: 2008-06-18

JDO 等。

View 层 本层是 JSF 的优势所在,在此层的技术主要包括:一组应用程序编程接口,它们用来表示 UI 组件,管理组件的状态,处理事件,验证输入,定义页面的导航和支持国际化;一个 JSP 标签库,它可以表示一个 JSP 页面里面的 JSF 界面。

Controller 层 JSF 的 Controller 层主要由一个 FacesServlet 类和一个或多个配置文件以及一系列的 action 处理器组成,它们共同控制 Web 应用程序的运作流程。

同以往的 Java Web 开发架构相比,JSF 的侧重点在 View 层,JSF 提供丰富的页面开发组件,而且支持组件的自定义开发,这使得它非常适用于 Web 页面复杂而业务逻辑简单的大、中型 Web 应用系统的开发。

3 应用

以用户登陆为例说明 JSF 程序的基本步骤,该例为:用户在页面输入用户名和密码,如输入正确,进入欢迎页面,若输入错误,则显示错误信息。

3.1 UserBean 定义

主要代码:

```

Public class UserBean ( )
{public String getName ( )
{return name ;}
public void setName ( String new Value)
{name = new Value ;}
public String Check( )
{if ( ! name. equals ("user") || ! password. equals
("123456") )
errMessage = "name or password is error ! ";
Return "failure"; else return "success";}}

```

其中函数 Check 的作用是验证用户输入的用户名和密码是否正确,判断其要进入下一个欢迎页面还是提示错误信息。

3.2 managed-bean 配置

设置 Bean 名称为 user, Bean 的生命周期为 session.

```

<managed-bean>
<managed-bean-name>user</managed-bean-name >
<managed-bean-class >com. jsf. UserBean </ managed-
bean-class>

```

```

<managed-bean-scope>session</managed-bean-scope>
</managed-bean>

```

3.3 页面导航设置

```

<faces-config>
<navigation-rule>
<from-view-id >/login.jsp </from-view-id >
<navigation-case>
<from-outcome>success</from-outcome>
<to-view-id >/welcome.jsp</to-view-id >
</navigation-case>
<navigation-case>
<from-outcome>failure</from-outcome>
<to-view-id >/login.jsp</to-view-id >
</navigation-case>
</navigation-rule>
</faces-config>

```

此例导航规则是:对于 login.jsp 页面,如果结果是"success",则显示 welcome.jsp 页面;如果结果是"failure",则重新显示 login.jsp 页面。

3.4 登录页面(login.jsp)及欢迎页面(welcome.jsp)

主要代码如下:

```

Login.jsp
<%@ taglib uri= "http://java.sun.com/jsf/core"
prefix= "f"%><!--引入核心标签库
<%@ taglib uri= "http://java.sun.com/jsf/html"
prefix= "h"%><!--引入 HTML 标签库
<f:view>
<h:form>
<h:inputText value= "# {user.name}"/ >
<h:inputText value= "# {user.password}"/ >
</h:form>
</f:view>
Welcome.jsp
<f:view >
Welcome to Java Server Faces
<h:outputText value= "# {user.name}"/ >
</f:view>

```

4 结语

JSF 是一种基于 MVC 设计模式的企业级 Web 应用开发框架,在基于 Java 的 Web 应用开发领域提供了一个可重用、可扩展、基于组件、工具友好的服务器端 UI 框架。JSF 降低了 Web 应用程序的开发难度,提高了生产效率。

参考文献

[1] Kito Mann. JSF 实战[M]. 铁手,程晓冬,何勇,译.北京:人民邮电出版社,2007: 12.
[2] 邝文清,郭跟成.基于 JSF 框架 Web 应用开发的研究[J]. 计算机应用研究,2007, 12(12): 273-275.
[3] 张猛,孙忠林.基于 MVC 模式的 JSF 架构的应用研究[J]. 福建电脑,2007(9): 108-109.

编辑:文心