浙江工业大学 JavaEE技术实验报告

实验名称:三Struts2 的控制器组件Action——登录用户的功能扩展

学院:计算机科学与技术学院、软件学院

班级： 软件工程1803

姓名： 石煌宽

组号： 无

时间： 2020/10/

**一、实验主要步骤**

**一、基础实验——Struts2 框架搭建**

（一）实验目的

1、掌握Struts2 的Action 类中自定义方法的使用；

2、掌握Struts2 中Action 类的不同调用方式和相应的配置方法；

3、掌握Action 的实例化情况，理解Action 与Servlet 在实例化上的区别；

4、理解JSP 文件中获取Action 属性的过程；

5、了解Struts2 支持的Action 处理结束后的结果类型。

（二）基本知识与原理

1、Action 类中的默认方法名是execute()方法，可以被自动调用；

2、在Action 中也允许定义其它方法名，可以同时定义多个方法，分别处理不同的逻辑；

3、当Action 中使用了自定义方法，则该Action 就需要特定的配置，一般有四种调用方式：

（1）在struts.xml 文件中通过method 属性指定方法名；

（2）使用动态方法调用方式（DMI）；

（3）使用提交按钮的method 属性；

（4）使用通配符配置Action；

4、Action 类是多实例的，Action 类的属性是线程安全的；

5、在JSP 页面中，可以通过Struts2 标签调用Action 中对应的getter 方法，从而输出Action 的属性值；

6、当一个Action 处理用户请求结束后，返回一个字符串作为逻辑视图名，再通过struts.xml 文件中的配置将逻辑视图名与物理视图资源关联起来；Struts2

默认提供了一系列的结果类型（struts-default.xml 配置文件的result-types 标签里列出了所支持的结果类型），结果类型决定了Action 处理结束后，将调

用哪种视图资源来呈现处理结果。

（三）实验内容及步骤

1、在IDEA中新建Web 工程struts-prj2，并将Struts2 中的8 个核心包添加到工程中；

2、在struts-prj2 中新建login.jsp 页面，作为用户登录的视图；新建loginFail.jsp页面，作为登录失败的视图；新建loginSuccess.jsp 页面，作为登录成功的视图（可重用“实验二Struts 基础应用”中基础实验里的页面代码）；

3、在struts-prj2 中新建register.jsp 页面，作为用户注册的视图；新建regFail.jsp

页面，作为注册失败的视图；新建regSuccess.jsp 页面，作为注册成功的视图（可重用“实验二Struts 基础应用”中提高实验里的页面代码）；

4、在struts-prj2 中新建cn.edu.zjut.bean 包，并在其中创建UserBean.java，用于记录用户信息（ 可重用“ 实验二Struts 基础应用” 中提高实验里的UserBean.java 代码）；

5、在struts-prj2 中新建cn.edu.zjut.service 包，并在其中创建UserService.java，用于实现登录逻辑和注册逻辑（可重用“实验二Struts 基础应用”中提高实验里的UserService.java 代码）；

6、在struts-prj2 中新建cn.edu.zjut.action 包，并在其中创建UserAction.java，定义login()方法和register()方法，参照“实验二Struts 基础应用”写入代码，分别用于调用登录逻辑和注册逻辑（代码片段如下）；

package cn.edu.zjut.action;

……

public class UserAction {

……

public String login() {

UserService userServ = new UserService();

if (userServ.login(loginUser)) {

return "success"; }

return "fail";

}

public String register() {

UserService userServ = new UserService();

if (userServ.register(loginUser)) {

return "registersuccess"; }

return "registerfail";

}

}

7、在工程struts-prj2 的src 目录中创建struts.xml 文件，用于配置Action 并设置页面导航，通过action 标签中method 属性指定方法名（代码片段如下）；

<struts>

<package name="strutsBean" extends="struts-default" namespace="/">

<action name="login" class="cn.edu.zjut.action.UserAction"

method="login">

<result name="success">/loginSuccess.jsp</result>

<result name="fail">/loginFail.jsp</result>

</action>

<action name="register" class="cn.edu.zjut.action.UserAction"

method="register">

<result name="registersuccess">/regSuccess.jsp</result>

<result name="registerfail">/regFail.jsp</result>

</action>

</package>

</struts>

8、编辑Web 应用的web.xml 文件，增加Struts2 核心Filter 的配置；

9、将struts-prj2 部署在Tomcat 服务器上，通过浏览器访问login.jsp 与register.jsp页面，并记录运行结果；

10、查找相关资料，尝试使用Action 自定义方法的其它三种调用和配置方式：

动态方法调用方式（DMI）、提交按钮的method 属性、通配符配置Action，

并记录关键配置和运行结果；

重新配置sturts.xml,如下

<action name="UserAction" class="cn.edu.zjut.action.UserAction">

<result name="success">/loginSuccess.jsp</result>

<result name="fail">/loginFail.jsp</result>

<result name="register\_success">/regSuccess.jsp</result>

<result name="register\_fail">/regFail.jsp</result>

</action>

1. 使用动态方法调用方式（DMI）；

配置方法：声明action的名字和类型，如

<action name="UserAction" class="...UserAction">

</action>

访问方式：以actionName!methodName.action形式访问，如

<s:form action="UserAction!login" method="post">

（2）使用提交按钮的method 属性；

配置方法：声明action的名字和类型，如

<action name="UserAction" class="...UserAction"></action>

访问方式：以actionName!methodName.action形式访问，如

<s:submit value="登录" method="login"/>

1. 使用通配符配置Action；

在name属性中使用通配符\*，method属性使用表达式{}，如

<action name="\*Action" class="...UserAction" method="{1}">

</action>

访问方式：以actionName.action形式访问，如

http://localhost:8080/struts2/loginAction.action

<http://localhost:8080/struts2/registerAction.action>

11、修改UserAction.java，增加UserAction 类的构造方法UserAction()，增加count属性，用于测试Action 的实例化情况（代码片段如下）；

public class UserAction {

private Integer count=0;

public UserAction(){

System.out.println("创建了一个 UserAction 类对象。");

}

public Integer getCount() {

return count;

}

public String login() {

count++; //Action 实例化情况测试

UserService userServ = new UserService();

if (userServ.login(loginUser)) {

return "success";

}

return "fail";

}……

}

12、修改 loginSuccess.jsp，在页面中使用标签输出 Action 中的 count 值；

13、重新将 struts-prj2 部署在 Tomcat 服务器上，通过浏览器访问 login.jsp 页面，并刷新多次，记录运行结果；

14、修改 struts.xml 文件，将 UserAction 的页面导航设置为 redirect 结果类型（代码片段如下）；

<struts>

<package name="strutsBean" extends="struts-default" namespace="/">

<action name="login" class="cn.edu.zjut.action.UserAction"

method="login">

<result name="success" type="redirect">

/loginSuccess.jsp</result>

<result name="fail">/loginFail.jsp</result>

</action>

</package>

</struts>

15、重新将 struts-prj2 部署在 Tomcat 服务器上，通过浏览器访问 login.jsp 页面，观察登录成功后 loginSuccess.jsp 页面的输出，并记录下来。

（四）实验要求

1、填写并上交实验报告，报告中应包括：

（1）运行结果截图；

（2）查找相关资料，根据实验过程，总结 Action 自定义方法的四种调用和配置方式；

（3）根据实验过程，记录实验步骤 11、12 中修改后的关键代码，及相应的运行结果或报错信息，分析 Action 的实例化情况，将 Action 与 Servlet 在实例化情况上进行对比，并记录下来；

（4）根据实验步骤 11、12，查找相关资料，分析 JSP 文件中获取 Action 属性的主要过程，并记录下来；

（5）根据实验步骤 11-15，观察两次 loginSuccess.jsp 页面输出上的区别，分析原因并记录下来；

（6）解压缩 Struts2 的核心包 struts2-core-2.3.15.1.jar，找到 struts-default.xml 配置文件，在其中的 result-types 标签里列出了 Struts2 所支持的结果类型，查找相关资料，总结这些结果类型的作用和特点，并记录下来；

（7）碰到的问题及解决方案或思考；

（8）实验收获及总结。

2、上交程序源代码，代码中应有相关注释。

**二、提高实验——ActionSupport 与输入校验**

（一）实验目的

1、了解 Action 接口的作用，理解 ActionSupport 类的作用；

2、掌握在 Struts2 中使用校验器或手工编码的方式，对请求参数进行数据校验的

方法，掌握在 JSP 页面中显示错误信息和提示信息的方法；

3、掌握在 Action 中使用国际化资源文件的方法；

4、掌握 Struts2 内置类型转换器的作用和使用方法。

（二）基本知识与原理

1、为了让用户开发的 Action 类更规范，Struts2 提供了一个 Action 接口，该接口定义了 Struts2 的 Action 处理类应该实现的规范；

2、Struts2 还为 Action 接口提供了一个实现类：ActionSupport，该类提供了若干默认方法，包括：默认的处理用户请求的方法（excute()方法）、数据校验的方法、添加校验错误信息的方法、获取国际化信息的方法等。

3、Struts2 框架提供了校验器和手工编码两种方式对请求参数进行数据校验，当 Action 类继承了 ActionSupport 类，就可以通过定义名为“<ActionAliasName>-validation.xml”的校验规则文件的方法进行校验器校验，也可以通过重写 ActionSupport 类的 validate()方法或 validateXxx()方法进行手动校验；

4、在 JSP 页面中使用 Struts2 标签生成的表单，能将域级别的错误信息将自动显示到表单元素处；

5、在 JSP 页面中使用 fielderror 标签，可以集中显示所有的域级错误信息；使用actionerror 标签，可以显示所有的 Action 级别错误信息；使用 actionmessage标签，可以显示 Action 消息；

6、Struts2 框架中提供了部分内置的类型转换器，可以将请求参数的 String 类型转换成基本数据类型及对应的包装器类型、日期类型、数组类型、集合类型等，当 Action 类继承了 ActionSupport 类，则内置的类型转换器将默认生效，可以直接使用；

7、如需修改默认的类型转换校验信息，则要在 Action 类的包中声明名为“Action类名.properties”的局部属性文件；

8、Struts2 框架同时支持自定义类型转换器，将请求参数转换成任意一种类型。

（三）实验内容及步骤

1、在struts-prj2中修改UserAction类，使其继承ActionSupport类，并在UserAction类中覆盖 ActionSupport 类的 validate()方法，用于对用户登录的请求参数account 和 password 进行校验：若用户名或密码为空，则使用 addFieldError（域级）添加错误信息（代码片段如下）；

import com.opensymphony.xwork2.ActionSupport; ……

public class UserAction extends ActionSupport { ……

public void validate() {

String account = loginUser.getAccount();

String pwd = loginUser.getPassword();

if (account == null || account.equals("")) {

this.addFieldError("loginUser.account", "请输入您的用户名！");

}

if (pwd == null || pwd.equals("")) {

this.addFieldError("loginUser.password", "请输入您的密码！");

}

}

}

2、修改 struts.xml 文件，在 Action 的配置中增加 validate()方法校验出错时的页面导航（<result name="input">），代码片段如下：

<struts>

<package name="strutsBean" extends="struts-default" namespace="/">

<action name="login" class="cn.edu.zjut.action.UserAction"

method="login">

<result name="success">/loginSuccess.jsp</result>

<result name="fail">/loginFail.jsp</result>

<result name="input">/login.jsp</result>

</action> ……

</package>

</struts>

3、重新将 struts-prj2 部署在 Tomcat 服务器上，通过浏览器访问 login.jsp 页面，观察并记录运行结果；

4、修改 login.jsp 页面，在表单前增加 fielderror 标签：<s:fielderror/>，再通过浏览器访问 login.jsp 页面，观察并记录运行结果；

5、修改 UserAction.java，在调用登录逻辑的 login()方法中，对登录情况进行校验：若登录成功，使用 addActionMessage()方法添加“登录成功！”的 Action 提示消息，若登录失败，使用 addActionError()方法添加 Action 级别的错误信息（代码片段如下）；

（四）实验要求

**三、扩展实验——Struts2 的国际化**

（一）实验目的

（二）基本知识与原理

（三）实验内容及步骤

（四）实验要求

**二、实验结果及问题分析**

**三、遇到的问题**

**四、实验总结及收获**