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；

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）；

（四）实验要求

**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“<ActionClassName>-<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！");

}i

f (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）；

public class UserAction extends ActionSupport {

……

public String login() {

UserService userServ = new UserService();

if (userServ.login(loginUser)) {

this.addActionMessage("登录成功！");

return "success";

} else {

this.addActionError("用户名或密码错误，请重新输入！");

return "fail";

}

}

6、修改login.jsp 页面，增加actionerror 标签（<s:actionerror/>）Action 级别的错误信息；修改loginSuccess.jsp，使用actionmessage 标签（<s:actionmessage/>）显示Action 提示消息；

7、修改struts.xml 文件中用户登录的页面导航设置，将登录失败时转向的页面从loginFail.jsp 修改为login.jsp；

8、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；

9、在工程struts-prj2 中创建“UserAction-login-validation.xml”校验规则文件，使其与UserAction 类位于同一目录下，配置校验信息，使用校验器对请求参数进行校验（代码如下）；

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE validators PUBLIC

"-//Apache Struts//XWork Validator 1.0.2//EN"

"http://struts.apache.org/dtds/xwork-validator-1.0.2.dtd">

<validators>

<field name="loginUser.account">

<field-validator type="requiredstring">

<param name="trim">true</param>

<message>用户名不能为空</message>

</field-validator>

</field>

<field name="loginUser.password">

<field-validator type="requiredstring">

<param name="trim">true</param>

<message>密码不能为空</message>

</field-validator>

</field>

</validators>

10、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；

11、参考“实验二Struts 基础应用”中的扩展实验，将login.jsp、loginSuccess.jsp、loginFail.jsp 三个页面进行国际化处理，把需要进行国际化的内容以键值对的形式写入资源文件message\_zh\_CN.properties 和message\_en\_US.properties中；

12、在资源文件中添加校验信息的键值对， 并使用native2ASCII 工具， 将

message\_zh\_CN.properties 重新编码，将中文字符都转化为unicode 码（代码

片段如下）；

#message\_en\_US.properties

#field error message

login.account.null=Please input your account!

login.password.null=Please input your password!

#action error message

login.error= Account or password error, please input again!

#action message

login.success=Login successfully!

（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**