浙江工业大学

JavaEE技术实验报告

实验名称：Struts2 基础应用——基于 Struts2 框架的用户登录模块

学 院：计算机院

班 级：软工1805

姓 名：张梦磊

学 号：201806062724

组 号：无

时 间：2020.10.14

一、基础实验——**Struts2** 框架搭建

# （一）实验目的

1. 掌握 Struts2 应用的基本开发步骤和常规配置；
2. 观察表单参数与 Action 属性的赋值关系，观察 Action 的 execute()方法及其返回值，并能够正确应用；
3. 观察配置文件 struts.xml 中的主要元素及属性，并能够正确应用；
4. 理解 Struts2 框架中 MVC 设计模式的体现，理解 Action，FilterDispatcher， struts.xml 的主要作用，并能够正确应用。

# （二）基本知识与原理

1. Struts2 是从 WebWork 框架上发展起来的 MVC 框架；
2. FilterDispatcher 是 Struts2 中的核心控制器，客户端对服务器端的请求将被 FilterDispatcher 过滤；若请求需要调用某个 Action，则框架将根据配置文件 struts.xml，找到需要调用的 Action 类；
3. Action 类是一个符合一定命名规范的 JavaSE 类，作为业务控制器使用；Action 中的 execute()方法用于调用 Model 层的业务逻辑类，并根据返回结果决定页面导航；
4. 若 Action 类中需要使用表单提交的请求参数，那么必须在 Action 类中声明与表单域的名字对应的变量，并为变量提供 getters/setters 方法；
5. Action 类需要在 struts.xml 中进行配置才能使用；
6. 编译运行基于 Struts2 框架的 Web 工程，需要导入 struts2 的 8 个核心 jar 包：表 2-1 strut2 的 8 个核心 jar 包

|  |  |
| --- | --- |
| 文件名 | 说明 |
| struts2-core-2.3.15.1.jar | Struts 2 框架的核心类库 |
| xwork-core-2.3.15.1.jar | XWork 类库，Struts 2 的构建基础 |
| Ognl-3.0.6.jar | Struts 2 使用的一种表达式语言类库 |
| freemarker-2.3.19.jar | Struts 2 的标签模板使用类库 |
| javassist-3.11.0.GA.jar | 代码生成工具包 |
| commons-lang3-3.1.jar | Apache 语言包，是 java.lang 包的扩展 |
| commons-io-2.0.1.jar | Apache IO 包 |
| commons-fileupload-1.3.jar | Struts 2 文件上传依赖包 |

# （三）实验内容及步骤

1. 登录 <http://struts.apache.org/download.cgi> 站点，下载 Struts2 的最新版（Full Distribution）；
2. 在 Eclipse 中新建 Web 工程 struts-prj1；
3. 将 Struts2 中的 8 个核心包增加到 Web 应用中，即复制到“%workspace% struts-prj1\WebContent\WEB-INF\lib”路径下，如下图所示；

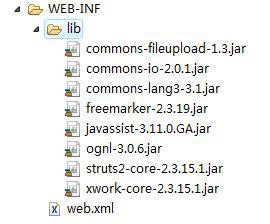


图 2-1 Struts2 的 8 个核心包

1. 在 struts-prj1 中新建 login.jsp 页面，作为用户登录的视图（代码片段如下），注意表单（form）中 action 属性的值，注意用户名与密码输入框中 name 属性的值；

|  |
| --- |
| <form action="login" method="post"> 请输入用户名：<input **name="loginUser.account"** type="text"><BR> 请输入密码：<input **name="loginUser.password"** type="password">  <input type="submit" value="登录"> </form> |

1. 在 struts-prj1 中新建 loginSuccess.jsp 和 loginFail.jsp 页面，分别作为登录成功或登录失败的视图，在页面中显示“登录成功”或“登录失败”；
2. 在 struts-prj1 中新建 cn.edu.zjut.bean 包，并在其中创建 UserBean.java，用于记录登录用户信息（代码如下），注意该 JavaBean 中属性名的写法；

package cn.edu.zjut.bean; public class UserBean { private **String account=""**;

|  |
| --- |
| private **String password=""**;  public String getAccount() { return account;  } public void setAccount(String account) { this.account = account;  } public String getPassword() { return password;  } public void setPassword(String password) { this.password = password;  }  } |

1. 在 struts-prj1 中新建 cn.edu.zjut.service 包，并在其中创建 UserService.java，用于实现登录逻辑，为简化登录逻辑，将登录成功的条件设置为：用户名和密码相同（代码如下）；

|  |
| --- |
| package cn.edu.zjut.service; import cn.edu.zjut.bean.UserBean; public class UserService { public boolean login(UserBean loginUser) { if(loginUser.getAccount().equals(loginUser.getPassword())){ return true;  } return false;  }  } |

1. 在 struts-prj1 中新建 cn.edu.zjut.action 包，并在其中创建 UserAction.java，调用登录逻辑，并根据登录结果不同而返回不同的内容（代码如下），注意该 Action 中的属性名及相应的 getters 和 setters 方法、execute()方法及返回值；

|  |
| --- |
| package cn.edu.zjut.action; import cn.edu.zjut.bean.UserBean; import cn.edu.zjut.service.UserService; public class UserAction { private **UserBean loginUser**; public UserBean **getLoginUser()** { return loginUser;  } public void **setLoginUser(UserBean loginUser)** { this.loginUser = loginUser;  } public **String** execute() { |
| UserService userServ = new UserService(); if (userServ.login(loginUser)) { **return "success"**;  } **return "fail"**;  }  } |

1. 在工程 struts-prj1 的 src 目录中创建 struts.xml 文件，用于配置 Action 并设置页面导航（代码片段如下），注意 action 标签中 name 属性和 class 属性的值，以及 result 子标签的属性；

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8" ?>  <!DOCTYPE struts PUBLIC  "-//Apache Software Foundation//DTD Struts Configuration 2.3//EN"  "http://struts.apache.org/dtds/struts-2.3.dtd">  <struts>  <package name="strutsBean" extends="struts-default" namespace="/">  <action **name="login" class="cn.edu.zjut.action.UserAction"**>  <result **name="success"**>/loginSuccess.jsp</result>  <result **name="fail"**>/loginFail.jsp</result> </action>  </package>  </struts> |

1. 编辑 Web 应用的 web.xml 文件，增加 Struts2 核心 Filter 的配置（代码片段如下）；

|  |
| --- |
| <!-- 定义Struts2的核心Filter -->  <filter>  <filter-name>struts2</filter-name>  <filter-class> org.apache.struts2.dispatcher.ng.filter  .StrutsPrepareAndExecuteFilter </filter-class>  </filter>  <!-- 让Struts2的核心Filter拦截所有请求 -->  <filter-mapping>  <filter-name>struts2</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping> |

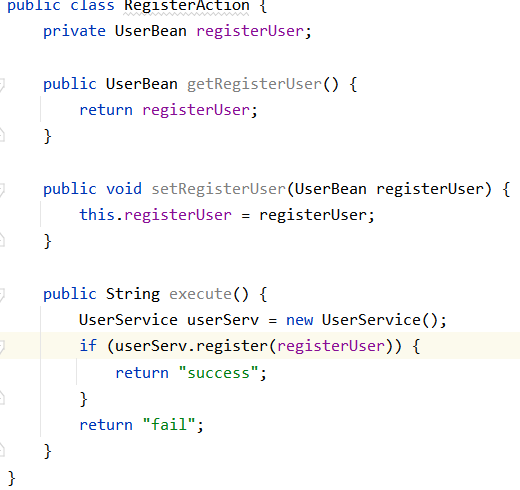
1. 将 struts-prj1 部署在 Tomcat 服务器上；
2. 通过浏览器访问 login.jsp 页面，并记录运行结果；
3. 尝试对以上第 4、6、8、9 中的关键代码进行修改，观察修改后的运行结果。

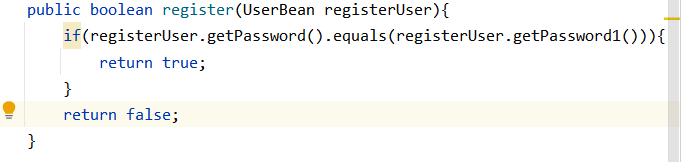
# （四）实验要求

1、填写并上交实验报告，报告中应包括：

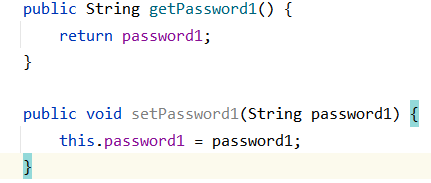
1. 运行结果截图；修改后的关键代码，及相应的运行结果或报错信息；

增加了注册功能

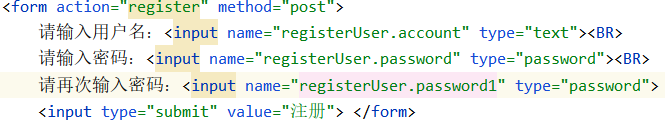




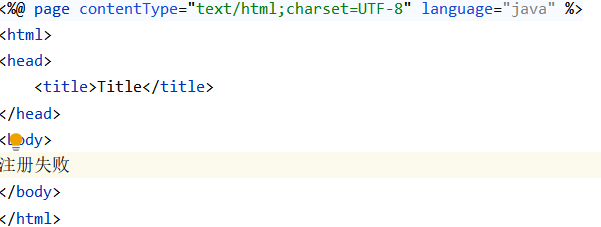
UserBean增加password1和相应的get，set

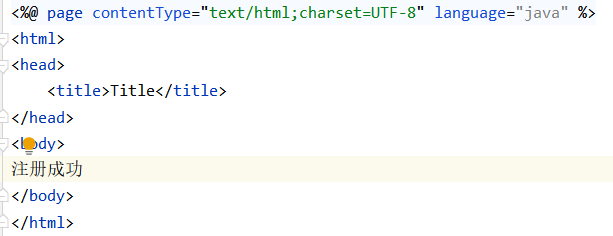


Register.jsp



两个结果视图





1. 根据实验过程，总结 jsp 页面、Action 类、Service 类、JavaBean、Filter 和 struts.xml 文件的作用，整理 Struts2 应用中从请求到响应的完整流程，思考并总结 Struts2 框架中 MVC 的体现；

1、jsp页面通常用作视图，用于提交表单数据及返回最后的结果。Javabean是数据模型，service是业务模型，两者配套使用。Action是控制器，可选用不同在模型实现不同的功能，它与提交数据的表单配套使用，当需要更改或者添加新功能时就不需要改变数据模型与业务模型，只需要在表单和action上做相应的更改就行。Struts.xml用于管理action请求映射。

表单提交数据，根据action的name去struts.xml文件中的找对应的action及Java类以确定action的访问路径，action自动执行excute方法根据返回的string去struts.xml中的result属性跳转视图。Action是控制器，控制页面流向，可调用service包中的业务模型及bean中的数据模型存储表单数据。

视图通常是jsp页面，用于提交表单数据及返回最后的结果。模型包括业务模型及数据模型。控制器包括action和struts.xml控制页面流向

1. 根据实验过程，总结表单参数与 Action 属性的赋值关系，并记录下来；

根据action属性中的class找到对应的java类，java类中的loginuser与表单中的loginuser是对应的。

1. 根据实验过程，总结 Action 的 execute()方法的作用和特点，并记录下来；

1、action默认执行execute()方法是因为继承了ActionSupport类，ActionSupport中的execute()方法是公共的String类，返回不同的string以便根据struts.xml中result属性跳转不同的视图

2、Action执行的时候并不一定要执行execute方法，即Action中不一定需要public String execute()方法，可以在配置文件中配置Action的时候用method=来指定执行哪个方法

1. 根据实验过程，查找相关资料，写出本实验中配置文件 struts.xml 里各元素及其属性的作用；

<action>标签

 代表配置action类，包含的属性

name              和<package>标签的namespace属性一起来决定访问路径的。

class                配置Action类的全路径（默认值是ActionSupport类）

 method          Action类中执行的方法，如果不指定，默认值是execute

## <result>标签

action类中方法执行，返回的结果跳转的页面

name      结果页面逻辑视图名称

<package>标签

代表的包的概念，将功能相似的action放在一起，包含的属性

name         包的名称，要求是唯一的，管理action配置。

extends      继承，可以继承其他的包，只要继承了，那么该包就包含了其他包的功能，一般都是继承struts-default

namespace  名称空间，一般与<action>标签中的name属性共同决定访问路径



1. 碰到的问题及解决方案或思考；
2. 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——**Struts2** 标签

# （一）实验目的

1. 进一步熟悉 Struts2 应用的基本开发步骤和常规配置；
2. 进一步熟悉 Action 及配置文件 struts.xml 的应用方法；
3. 掌握 Struts2 标签的基本使用方法；
4. 能熟练使用 Struts2 的常用标签，能参考 Struts2 标签的使用说明文档，对各类标签进行灵活应用；

# （二）基本知识与原理

1. 使用 Struts2 标签的形式来表达页面逻辑，可以尽量避免在视图中使用 Java 代码，让逻辑与显示分离，提高视图的可维护性；
2. Struts2 标签库的主要 tld 文件为 struts-tags.tld，在 struts2-core-2.3.15.1.jar 包中，另一个与 Ajax 相关的标签库 tld 文件为 struts-dojo-tags.tld，在 struts2-dojo

-plugin-2.3.15.1.jar 包中；

1. Struts2 标签的使用步骤和使用 JSTL 相同，只需在 JSP 页面中使用 taglib 指令引入标签库中 tld 文件的 uri，并指定前缀即可，例如：

<%@ taglib prefix="s" uri="/struts-tags"%>；

1. 根据 Struts2 标签的主要作用，可以将其分为：用于生成页面元素的 UI 标签、用于实现流程控制的控制类标签、用于控制数据的数据标签和用于支持 Ajax 的标签。

# （三）实验内容及步骤

1. 为能使用与 Ajax 相关的标签，将 Struts2 中的 struts2-dojo-plugin-2.3.15.1.jar 包增加到工程 struts-prj1 中，即复制到“%workspace%struts-prj1\WebContent\ WEB-INF\lib”路径下，刷新工程；
2. 在工程 struts-prj1 中增加用户注册功能，新建 register.jsp 页面作为用户注册的视图，页面使用 Struts2 的 UI 标签来生成表单元素，包括用户名、密码、确认密码、真实姓名、性别、生日、联系地址、联系电话和电子邮箱（代码片段如下）；

|  |
| --- |
| **<%@ taglib prefix="s" uri="/struts-tags"%>**  **<%@ taglib prefix="sx" uri="/struts-dojo-tags"%>**  <html>  <head>  <s:head theme=*"xhtml"*/>  <sx:head parseContent=*"true"* extraLocales=*"UTF-8"*/> </head>  <body>  **<s:form action="register" method="post">**  <s:textfield name="loginUser.account" label="请输入用户名"/>  <s:password name="loginUser.password" label="请输入密码"/>  ……  <s:radio name="loginUser.sex" list="#{1 : '男', 0 : '女'}" label  ="请输入性别"/>  <sx:datetimepicker name="loginUser.birthday" displayFormat  ="yyyy-MM-dd" label="请输入生日"/>  ……  <s:submit value="注册"/>  <s:reset value="重置"/>  **</s:form>**  </body>  </html> |

1. 在 struts-prj1 中新建 regFail.jsp 页面，作为注册失败的视图，在页面中显示“注册失败”；
2. 在 struts-prj1 中新建 regSuccess.jsp 页面，作为注册成功的视图，使用 Struts2 的数据标签和控制标签来生成注册成功的信息，并将登录用户信息保存在会话范围内（代码片段如下）；

<!-- 数据标签property -->

<s:property value="loginUser.name"/>

<!-- 控制标签if/else -->

|  |
| --- |
| <s:if test="%{loginUser.sex==\"1\"}">  <s:text name="先生，"/>  </s:if>  <s:else>  <s:text name="女士，"/>  </s:else>  您注册成功了！  <!-- 数据标签set -->  <s:set name="user" value="loginUser" scope="session"/> |

1. 修改 UserBean.java，增加属性用于记录注册用户信息（代码片段如下）；

|  |
| --- |
| public class UserBean { private String account=""; private String password=""; private String repassword=""; private String name=""; private String sex=""; private String birthday=""; private String address=""; private String phone=""; private String email="";  //省略getters/setters方法 } |

1. 修改 UserService.java，增加用户注册逻辑，为简化注册逻辑，将注册成功的条件设置为：用户名、密码和确认密码相同，而且不为空字串；
2. 修改 UserAction.java 中的 execute()方法，参照基础实验部分写入代码，用于调用注册逻辑，并根据注册成功与否而返回不同的内容；
3. 修改 struts.xml 文件，对用户注册进行配置并设置页面导航；
4. 将 struts-prj1 重新布署在 Tomcat 服务器上，通过浏览器访问 register.jsp 页面，并记录运行结果；
5. 查找 Struts2 标签的相关资料，并尝试将其它的 Struts2 标签应用在实验步骤

2、3、4 中，观察应用后的运行结果。

# （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. 应用各种 Struts2 标签的关键代码，及相应的运行结果或报错信息；
3. 根据实验过程，查找相关资料，总结 Struts2 中标签及其属性的作用和用法并记录下来；

控制标签类型

If /elseif /else 控制选择输出标签

append用于将多个集合并成一个新集合

generator将一个字符串解析成集合

iterator用于将集合迭代输出

merge将多个集合并成一个新集合

sort对集合进行排序

subset截取结合元素，形成新的集

iterator标签

对集合进行迭代，集合包含list,set,数组,map类型对象进行输出,属性：

status制定迭代时的IteratorStatus实例,通过该实例即可判断当前迭代元素的属性

value指定迭代的集合,若没有指定则使用ValueStack栈顶的集合

id指定集合里元素的id

generator标签

将指定字符串按指定分隔符分割成多个子字符串,属性：

count生成集合元素的总数

Seperator，指定分隔符

Val，指定被解析的字符串

Converter负责将集合中的每个字符串换成对象

Id将生成的集合放在request属性中

数据标签类型

action 直接调用一个action,并且可以获得执行返回的结果

Set设置一个新变量，将变量放入指定范围

param用于设置参数

property输出valueStack，stackContent，ActionContext中的值

I18n指定国际化资源

bean用于创建一个java对象

date格式化输出一个日期

url用于生成url地址

debug生成调试链接

include包含其他jsp

Push将某个值放入valueStack

property标签

用于输出指定值

default：可选属性，如果需要输出的属性值为null，则显示该属性指定的值

value：可选属性，指定需要输出的属性值，如果没有指定该属性，则默认输出ValueStack栈顶的值

id：可选属性，指定该元素的标识

1. 碰到的问题及解决方案或思考；
2. 实验收获及总结。

2、上交程序源代码，代码中应有相关注释。

三、扩展实验——**Struts2** 的国际化

# （一）实验目的

1. 进一步熟悉 Struts2 标签的基本使用方法；
2. 能使用 Struts2 标签实现国际化；
3. 了解配置文件 struts.properties 以及国际化资源文件的作用和基本使用方法。

# （二）基本知识与原理

1. 通过将不同语言版本的字符保存在属性文件中，Struts2 的国际化机制能够在不需要修改程序主体的前提下，实现不同语言版本的应用；
2. 在 Web 应用中选择需要进行国际化的内容，不在页面中直接输出该信息，而是通过 Struts2 标签输出一个键值，该键值在不同语言环境下对应不同的字符串；如：<s:textfield name="loginUser.account" lable="请输入用户名"/>代码中的“请输入用户名”是需要国际化的内容，将其以键值代替为<s:textfield name="loginUser.account" key="login.account.lable"/>；
3. 需要进行国际化的内容以键值对（key=value）的形式写入 Struts2 的国际化资源文件中，如“login.account.lable=请输入用户名”；该资源文件名可以自定义，但是后缀必须是 properties；资源文件应放在 Web 应用的类加载路径下；每一个语言版本需要创建一个资源文件；
4. 通过 Struts2 的配置文件 struts.properties 来配置资源文件的基础名，若资源文件的基础名为 message，则 message\_zh\_CN.properties 是对应的中文资源文件， message\_en\_US.properties 是对应的美国英语资源文件；
5. 浏览器将根据其默认的语言版本，自动调用相应语言的资源文件，从而在页面中展示不同的语言效果。

# （三）实验内容及步骤

1、在工程 struts-prj1 的 src 目录下新建一个 cn.edu.zjut.local 包，把所有的资源文件放置其中，如：message\_zh\_CN.properties、message\_en\_US.properties 等； 2、将工程 struts-prj1 中用户登录模块的三个 JSP 页面进行国际化处理，选择需要进行国际化的内容，以键值对的形式写入资源文件中（代码片段如下）；

# message\_en\_US.properties login.account.lable=Please input your account login.password.lable=Please input your password login.submit.button=submit

# message\_zh\_CN.properties login.account.lable=请输入用户名 login.password.lable=请输入密码 login.submit.button=登录

1. 使用 JDK 中的 native2ASCII 工具，将 message\_zh\_CN.properties 重新编码，将中文字符都转化为 unicode 码，从而避免乱码问题；

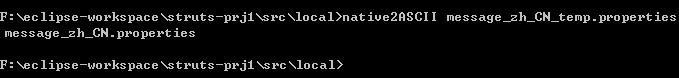


图 2-2 使用 native2ASCII 工具

# message\_zh\_CN.properties login.account.lable=\u8bf7\u8f93\u5165\u7528\u6237\u540d login.password.lable=\u8bf7\u8f93\u5165\u5bc6\u7801 login.submit.button=\u767b\u5f55

1. 在工程 struts-prj1 的 src 目录中创建 struts.properties 文件，通过它加载资源文件（具体代码如下）；

**struts.custom.i18n.resources=cn.edu.zjut.local.message** struts.i18n.encoding=GBK

或者在JSP页面中临时加载资源文件（代码片段如下）；

|  |  |
| --- | --- |
| **<s:i18n name="cn.edu.zjut.local.message">**  <s:form action="login" method="post"> …… **</s:i18n>** | </s:form> |

1. 修改 login.jsp、loginSuccess.jsp 和 loginFail.jsp 页面，通过 Struts2 标签实现国际化（代码片段如下）；

<!-- login.jsp -->

<s:textfield name="loginUser.account" **key="login.account.lable"**/> <s:password name="loginUser.password" **key="login.password.lable"**/>

<s:submit name="submit" **key="login.submit.button"**/>

1. 将 struts-prj1 重新布署在 Tomcat 服务器上；
2. 设置浏览器的首语言选项（如下图所示），通过浏览器访问 login.jsp 页面，观察并记录运行结果；



图 2-3 浏览器的语言首选项

1. 尝试修改用户注册模块，实现国际化，并记录运行结果。

# （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. 应用国际化的关键代码，及相应的运行结果或报错信息；

register.account.label=Please input your account  
register.password.label=Please input your password  
register.repassword.label=Please input your again  
register.name.label=Please input your name  
register.sex.label=Please input your sex  
register.birthday.label=Please input your birthday  
register.address.label=Please input your address  
register.phone.label=Please input your phone  
register.email.label=Please input your email  
register.sex.man.label=Man  
register.sex.woman.label=Woman  
register.submit.button=Register  
register.reset.button=Reset  
register.result1=Register success  
register.result2=Register fail  
register.man.label=gentleman  
register.woman.label=lady

register.account.label=请输入用户名  
register.password.label=请输入密码  
register.repassword.label=请再次输入密码  
register.name.label=请输入姓名  
register.sex.label=请输入性别  
register.birthday.label=请输入你的生日  
register.address.label=请输入你的地址  
register.phone.label=请输入你的电话  
register.email.label=请输入你的邮箱  
register.sex.man.label=男  
register.sex.woman.label=女  
register.submit.button=注册  
register.reset.button=重置  
register.result1=注册成功  
register.result2=注册失败  
register.man.label=先生  
register.woman.label=女士

1. 查找相关资料，总结配置文件 struts.properties 的作用并记录下来；
2. 碰到的问题及解决方案或思考；
3. 实验收获及总结。
4. 上交程序源代码，代码中应有相关注释。

出现的问题：

应用国际化标签后页面空白



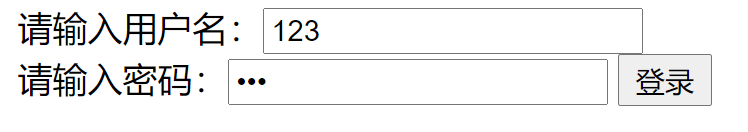


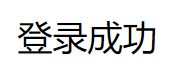
解决：页面上加上<%@ taglib prefix="s" uri="/struts-tags"%>  
<%@ taglib prefix="sx" uri="/struts-dojo-tags"%>后正常显示，但error依旧存在，可能是gradle构建项目的原因

1. 实验结果及问题分析

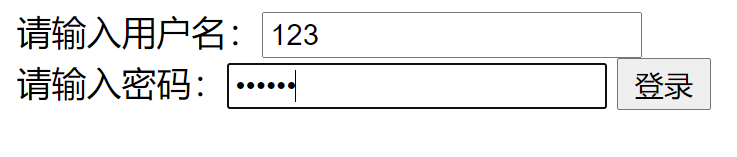
实验一运行结果：

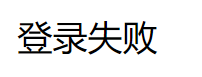
用户名密码一致时登录成功





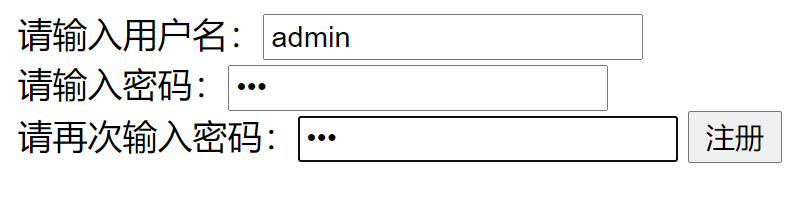
用户名密码不一致时登录失败

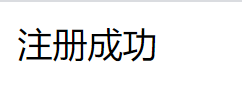




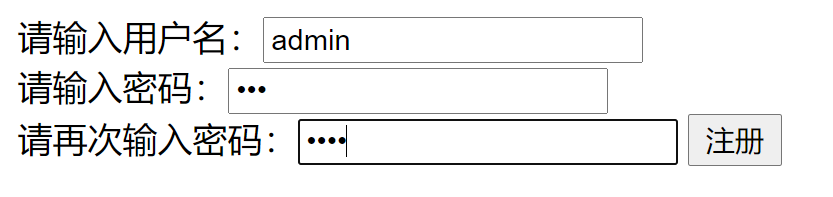
增加注册功能：

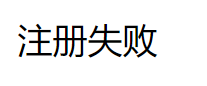
当两次密码一致时注册成功





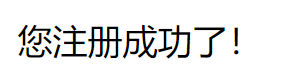
当两次密码不一致时注册失败



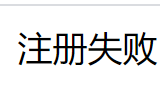


实验二运行结果：







实验三运行结果

