

XI`AN TECHNOLOGICAL UNIVERSITY

实验报告

实验课程名称 组件技术

专 业： 软件工程

班 级： 17060721

姓 名： 李正阳

学 号： 17060208112

实验学时： 2

指导教师： 白小军

成 绩：

2019 年 11 月 11 日

西安工业大学实验报告

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 专业 | 软件工程 | 班级 | 17060721 | 姓名 | 李正阳 | 学号 | 17060208112 |
| 实验课程 | 组件技术 | 指导教师 | 白小军 | 实验日期 | 2019/11/11 | 同实验者 | 无 |
| 实验项目 | JSP技术基础 | | | | | | |
| 实验设备及器材 | PC机 | | | | | | |

1. 实验目的

1）理解JSP的基本工作原理及执行过程；

2）熟悉JSP中常用的页面元素，理解其基本用法；

3）熟悉JSP的主要内置对象，掌握5个重要内置对象的用法；

4）学会使用JSP技术开发动态网页，构建简单的Web应用

二、实验原理

1、JSP的基本工作原理及执行过程，JSP与Servlet之间的关系；

JSP是Servlet的一种，主要是为了在HTML中混入Java代码能让网页 拥有动态的功能，客户端访问jsp文件，也是发出http请求之后，服务器 容器将jsp文件转为servlet。

1. JSP中的常用页面元素及其用法；

静态内容：和HTML语法一样

表达式：<%= 表达式 %> 在JSP中显示定义的变量

脚本：在JSP中执行java代码格式如下：

<% java代码 %>

声明：在JSP中定义变量或者方法

注释：HTML注释<!-- -->（客户端可见）

Jsp注释<% -- -- %>（客户端不可见）

1. JSP中的5个重要内置对象及其用法；

一共是九个内置对象：

（1）out 对象：负责管理对客户端的输出  
 （2）request 对象：负责得到客户端的请求信息  
 （3）response 对象：负责向客户端发送响应  
 （4）session 对象：负责保存同一客户端一次会话过程中的一些信息  
 （5）application对象 ： 表示整个应用的环境信息  
 （6）exception 对象：表示页面上发生的异常，可以通过它获得页面 异常信息  
 （7）page 对象：表示的是当前jsp页面本身，就像java类中定义的 this一样  
 （8）pageContext 对象：表示的是此jsp的上下文  
 （9）config对象：表示此jsp的ServletConfig

（1）out对象可以向客户端发送文本数据，所以有常用的以下两个方法：

void print();

void println();

（2）request对象主要获取客户端发送的数据：

主要是获取表单提交上来的数据：

Object getParameter();

（3）response对象主要是用来跳转：

sendRedirect();方法用来重定向

原理是让客户端去请求资源，所以无法带着数据跳转

forward()方法表示服务器去请求资源，可以带着数据跳转

（4）session负责保存同一客户端一次会话过程中的一些信息

主要用法就是将某些数据存储在Session中，例如购物车或者验证

1. application可以被多个客户端访问同一个对象

Object application.getAttribute(String name);

1. 实验步骤、数据记录及处理
2. 创建用户注册页面，并使用JSP技术实现后台处理，将注册信息保存到数据库，在注册成功后自动跳转到登录页面；

在JSP中实现后台处理代码registerservlet.jsp:

<%@ **page import**="**com.darksouls.dao.UserDao**" %>  
<%@ **page import**="**com.darksouls.dao.UserDaoImpl**" %>  
<%@ **page import**="**com.darksouls.vo.User**" %>*<%--  
 Created by IntelliJ IDEA.  
 User: 李正阳  
 Date: 2019/11/24  
 Time: 15:17  
 To change this template use File | Settings | File Templates.  
--%>*<%@ **page contentType**="**text/html;charset=UTF-8**" **language**="**java**" %>  
<**html**>  
<**head**>  
 **<%** request.setCharacterEncoding(**"UTF-8"**);  
 response.setCharacterEncoding(**"UTF-8"**);  
 String username = request.getParameter(**"name"**);  
 String userpassword = request.getParameter(**"password"**);  
 String email = request.getParameter(**"mail"**);  
 UserDao userDao = **new** UserDaoImpl();  
 **if**(userDao.SelectUser(username,userpassword) == 0){  
 User user = **new** User(username,userpassword,email);  
 userDao.InsertUser(user);  
 request.getRequestDispatcher(**"../WEB-INF/view/success.jsp"**).forward(request,response);  
 }**else** {  
 request.getRequestDispatcher(**"../WEB-INF/view/error.jsp"**).forward(request,response);  
 }  
 **%>** <**title**>Title</**title**>  
</**head**>  
<**body**>  
</**body**>  
</**html**>

N$`S}7{~GJ~({MYUNT`8A]W

点击注册按钮直接跳转到处理功能的jsp，处理完成后直接跳转到登陆界面

并且数据库出现记录：

{6MI7[V~CHZ_WQDBNVNWL9R

如果已经注册过就跳转为错误界面。

1. 创建用户登录页面，并使用JSP技术实现后台处理，查询数据库，验证用户名和密码，根据验证结果跳转到成功页面或失败页面；

使用jsp技术改写以前Servlet技术的登陆功能，loginservlet.jsp如下：

<%@ **page import**="**java.util.HashMap**" %>  
<%@ **page import**="**com.darksouls.dao.UserDao**" %>  
<%@ **page import**="**com.darksouls.dao.UserDaoImpl**" %>  
<%@ **page import**="**com.alibaba.druid.util.StringUtils**" %>*<%--  
 Created by IntelliJ IDEA.  
 User: 李正阳  
 Date: 2019/11/24  
 Time: 16:31  
 To change this template use File | Settings | File Templates.  
--%>*<%@ **page contentType**="**text/html;charset=UTF-8**" **language**="**java**" %>  
<**html**>  
<**head**>  
 <**title**>Title</**title**>  
</**head**>  
<**body**>  
**<%!  
 int init** = 0;  
**%>  
<%** request.setCharacterEncoding(**"utf-8"**);  
 response.setCharacterEncoding(**"utf-8"**);  
 String userName = request.getParameter(**"name"**);  
 String userPassword = request.getParameter(**"password"**);  
 UserDao userDao = **new** UserDaoImpl();  
 **try** {  
 **int** n = userDao.SelectUser(userName,userPassword);  
 */\*\*  
 \* 验证登陆  
 \*/* **if**(!StringUtils.*isEmpty*(userName)){  
 **if**(n == 1){  
 */\*\*  
 \* 添加session，以及访问次数  
 \*/* **init**++;  
 **if**(session.getAttribute(userName) !=**null**){  
 **int** num = (**int**)session.getAttribute(userName);  
 session.setAttribute(userName,num);  
 }**else** {  
 session.setAttribute(userName,**init**);  
 }  
 request.getSession().setAttribute(**"USER\_NAME"**,userName);  
 request.getSession().setAttribute(**"NUM"**,session.getAttribute(userName));  
 request.getRequestDispatcher(**"../WEB-INF/view/success.jsp"**).forward(request,response);  
 }**else** {  
 request.setAttribute(**"ERROR\_INFO"**,**"<span style='color:red;'>用户名密码错误！</span>"**);  
 request.getRequestDispatcher(**"../index.jsp"**).forward(request,response);  
 }  
 }**else** {  
 request.setAttribute(**"ERROR\_INFO"**,**"<span style='color:red;'>用户名密码错误！</span>"**);  
 request.getRequestDispatcher(**"../index.jsp"**).forward(request,response);  
 }  
 }**catch** (Exception e){  
 e.printStackTrace();  
 }  
*// request.getRequestDispatcher("WEB-INF/view/error.jsp").forward(request,response);***%>**</**body**>  
</**html**>

同时有验证账号是否为空以及记录下登陆的次数放在session中。

1. 使用JSP技术改造留言本应用，用JSP实现添加留言、显示所有留言等核心处理功能；

添加留言：messagecommit.jsp 将留言的内容时间和标题与用户的id号关联起来,形成了一对多的关系

<%@ **page import**="**com.darksouls.dao.UserDao**" %>  
<%@ **page import**="**com.darksouls.dao.UserDaoImpl**" %>  
<%@ **page import**="**java.sql.Date**" %>  
<%@ **page import**="**java.text.DateFormat**" %>  
<%@ **page import**="**java.text.SimpleDateFormat**" %>*<%--  
 Created by IntelliJ IDEA.  
 User: 李正阳  
 Date: 2019/11/24  
 Time: 18:05  
 To change this template use File | Settings | File Templates.  
--%>*<%@ **page contentType**="**text/html;charset=UTF-8**" **language**="**java**" %>  
<**html**>  
<**head**>  
 <**title**>Title</**title**>  
</**head**>  
<**body**>  
<**jsp:include page="checkUserLevel.jsp"**></**jsp:include**>  
**<%** String message = request.getParameter(**"message"**);  
 String usename = (String)request.getSession().getAttribute(**"USER\_NAME"**);  
 String title = request.getParameter(**"title"**);  
 UserDao userDao = **new** UserDaoImpl();  
 **int** id = userDao.selectUserId(usename);  
 java.util.Date utilDate = **new** java.util.Date();  
 Date sqlDate = **new** Date(utilDate.getTime());  
 userDao.insertMessage(message,id,title,sqlDate);  
 request.getRequestDispatcher(**"../WEB-INF/view/success.jsp"**).forward(request,response);  
**%>**</**body**>  
</**html**>

界面如下图：



查看留言的界面如下：



1. 为留言本应用添加访问控制功能，只有登录用户才能查看留言；当未登录用户请求留言列表页面时，自动跳转到登录页面。

首先明确WEB-INF中的文件不可以直接访问，只能够跳转所以我已经设计好了，只有登录用户才能查看的要求，但是为了满足题意，我在外面只能再创建两个文件用来演示效果。

关于权限检查的jsp文件：

CheckUserLevel.jsp：

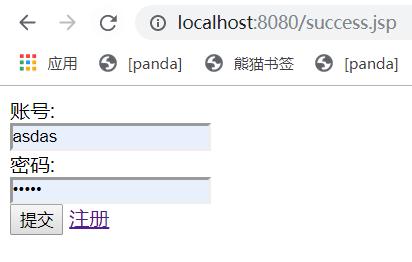
原理是查询服务器中session对象是否保存了关于这个用户的姓名，在登陆中会将姓名保存到session。

*<%--  
 Created by IntelliJ IDEA.  
 User: 李正阳  
 Date: 2019/12/1  
 Time: 19:20  
 To change this template use File | Settings | File Templates.  
--%>*<%@ **page contentType**="**text/html;charset=UTF-8**" **language**="**java**" %>  
<**html**>  
<**head**>  
 <**title**>Title</**title**>  
</**head**>  
<**body**>  
 **<%** request.setCharacterEncoding(**"utf-8"**);  
 response.setCharacterEncoding(**"utf-8"**);  
 String name = (String) request.getSession().getAttribute(**"USER\_NAME"**);  
 **if**(name == **null**){  
 request.getRequestDispatcher(**"login.jsp"**).forward(request,response);  
 }  
 **%>**</**body**>  
</**html**>

同时在留言板面板上还需要导入权限控制的文件代码如下图：

<**jsp:include page="checkUserLevel.jsp"**></**jsp:include**>

结果如下图,可见请求了success.jsp成功登陆界面，检查未登陆就跳转到了登陆界面。



1. 为留言本应用增加访问计数功能，每当有用户访问留言列表页面时，为访问计数值加1，并在页面上显示到目前总共有多少次用户访问留言页面。

本功能可以采用application对象进行设计。

功能实现在success.jsp中，代码片段如下所示：

**<%!  
 private int visit** = 0;  
**%>**

**<%** application.setAttribute(**"VISIT"**,**visit**);  
**%>**本页面一共被访问了**${**VISIT**}**次

结果如图所示：

