**实习一 JSP 技术**

**动态网页与静态网页最根本的区别是：动态网页中被嵌入服务器端脚本代码，Web服务器会根据客户端的请求去执行这些代码，并把执行的结果和页面中的静态内容一起发送给客户端。**

**交互式网页的特点：**

1. **具有良好的交互性，可以根据不同的要求动态产生页面内容**
2. **支持后台数据库，可以将数据永久地保存在数据库，有利于数据的查找**
3. **使用动态页面，页面量减少，降低了网页的维护工作量，提高了工作效率**

**实习要求**

学习使用 MyEclipse 集成开发环境下创建 Javav Web 应用，使用 JSP 内置对象进行编写交互式网页。

**实习目的**

掌握 MyEclipse 中 Web 应用的开发以及在 Tomcat 中的部署和访问。

掌握 JSP 常用内置对象的使用。

**实习环境**

服务器：Tomcat6 或更高

集成开发环境：MyEclipse2014 或更高

**实习内容**

# 1、MyEclipse 开发环境配置

在做 Web 开发前必须配置开发环境编码及 JSP 页面编码并保持一致，在本课程实习过程中所有编码都设置为 UTF-8。

# 配置开发环境的编码

点击菜单“window”->“preferences”在弹出对话框的左侧栏选择“General”->“WorkSpace” 如图 1。

# 配置 JSP 页面的编码

仍然在图 1 窗口的左侧栏选择“MyEclipse”-> “ Files and Editors”->“JSP”，在右侧的 “Creating files”的 Encoding 中指定 JSP 的编码如图 2（UTF-8）。最后点击“OK”按钮。、

# 配置服务器（Web 运行环境）

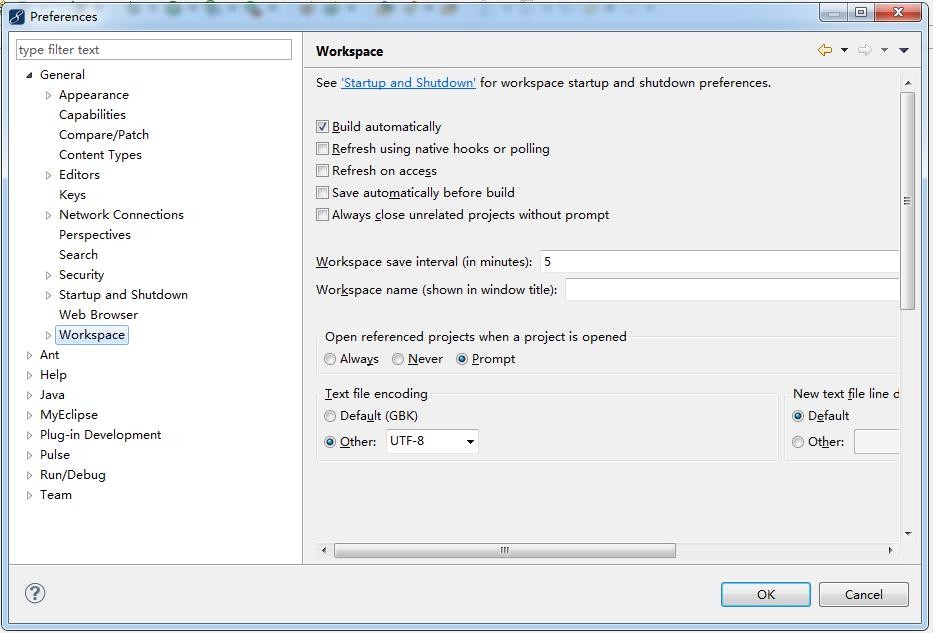


图 1 开发环境编码设置

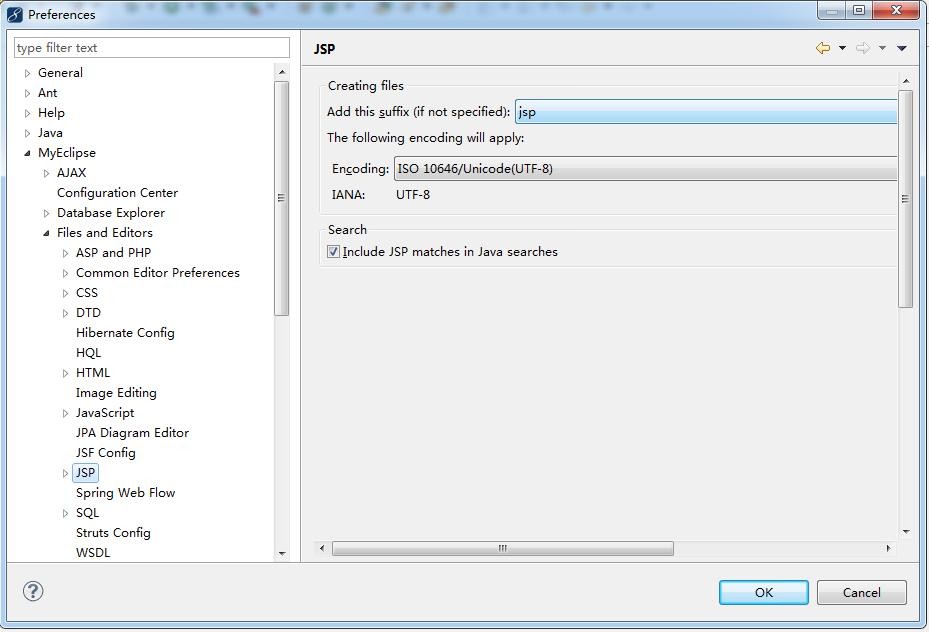


图 2 JSP 页面编码设置

MyEclipse 已经内置了一个Web 服务器——Tomcat，在 MyEclipse 窗口的下方选择“Servers”选项卡（若没有点击菜单“Window”->“Show Views”->“Servers”打开即可），如图 3。若

想使用其他的 Web 服务器，可点击图 3 右上侧的 （configure Server Connector），打开如图 4 窗口。在左侧选择待配置的服务器，以 Tomcat7 为例，选择左侧的“Tomcat”->“Tomcat7.x”，

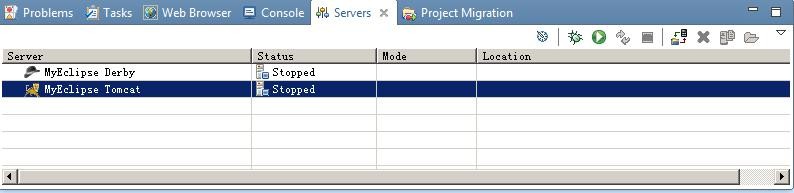


图 3 Servers 选项卡

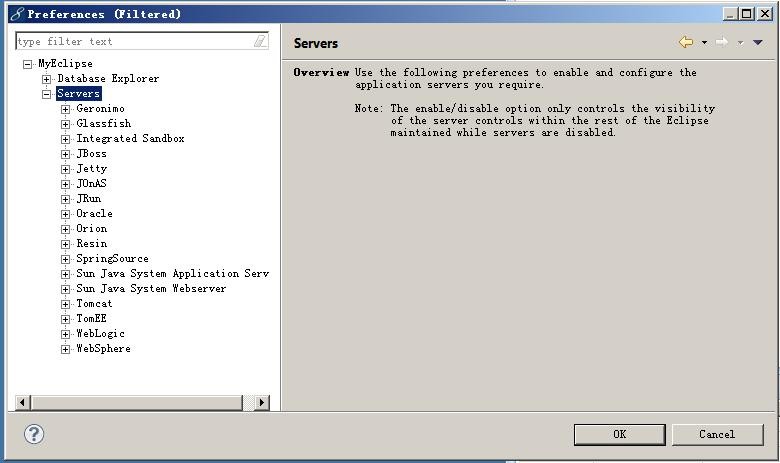


图 4 Web 服务器配置窗口

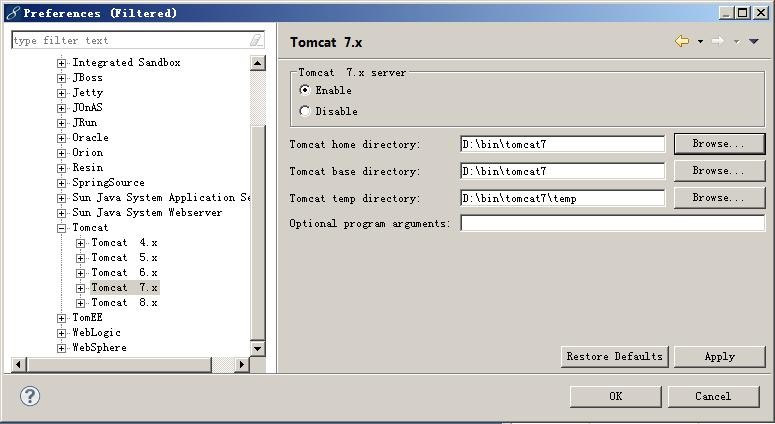
然后在右侧选择“Enable”并在下方指定 Tomcat7 的安装路径，最后点击“OK”按钮即可， 如图 5。

图 5 配置 Tomcat7.x

配置完成后，在 Servers 选项卡中便可显示已配置好的 Tomat7，如图 6。2、第一个 Web 应用

1. 创建 Web 应用

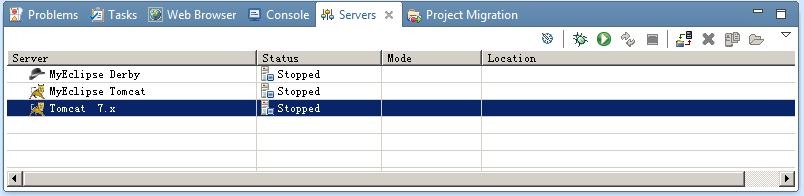


图 6 配置好的 Tomcat7.x

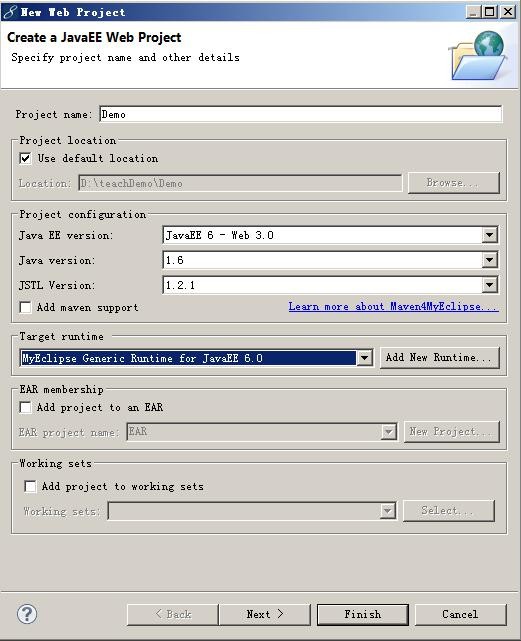
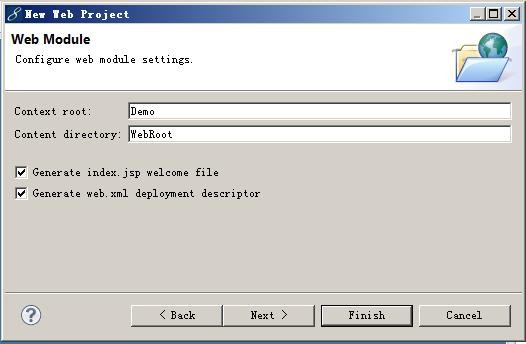
点击菜单“File”->“New”->“Web Project”或工具栏行的“”按钮->“Web Project”，弹出如图 7“New Web Project”向导。输入“Project name”（工程名，默认为应用的上下文）为 Demo，其他使用默认即可，点击“Next”，在下一个页面也点击“Next”，显示图 8 对话框，勾选第二个复选框（用于生产应用配置文件web.xml），最后点击“Finish”按钮完成创建。创建完成的工程结构如图 9。其中“src”用于存放 Java 源码，“WebRoot”存放网页相关文件： Html 文件、图片文件、JS、CSS、JSP 页面文件等。index.jsp 为向导创建的默认首页。

图 8 向导 2

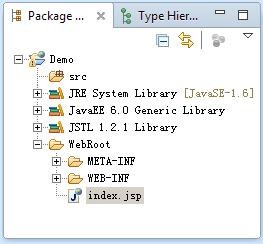


图 7 创建 Web 应用向导

1. 编辑 JSP 页面

图 9 Web 应用工程结构

双击打开 index.jsp，如图 10 JSP 编辑页面；上方为页面预览页面，下方为源码编辑框。自行修改页面中要显示的内容，完成后保存即可。

1. 部署应用

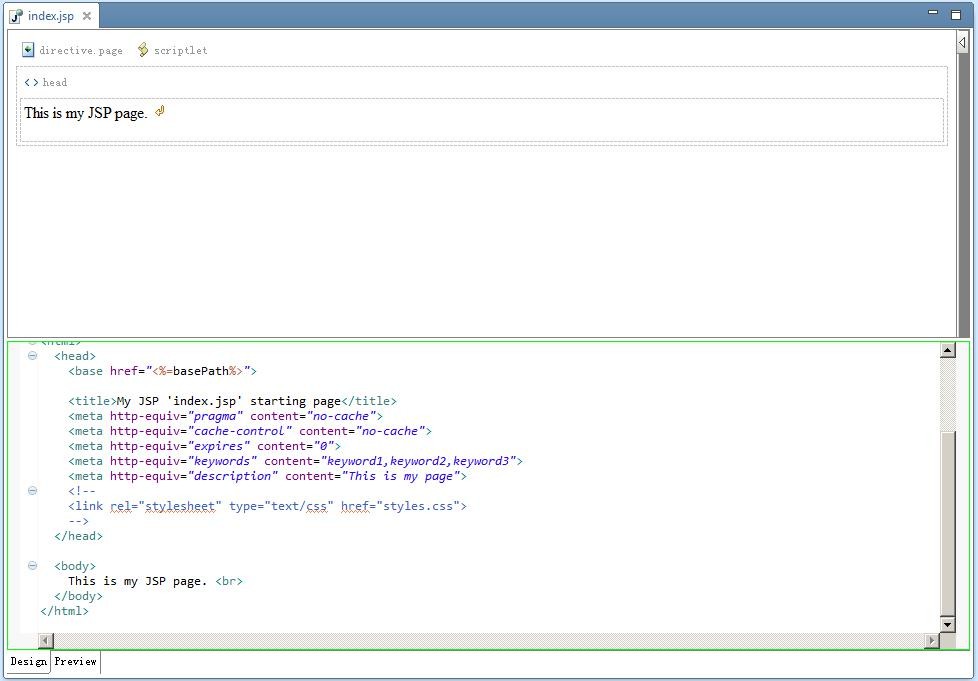
点击工具栏中的“ ”按钮，弹出如图 11 窗口。点击“Add”按钮，弹出如图 12 对话

图 10 JSP 编辑界面

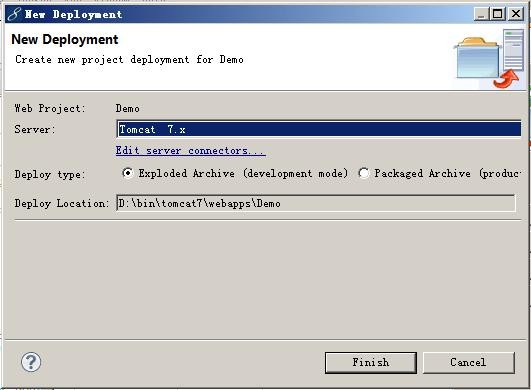
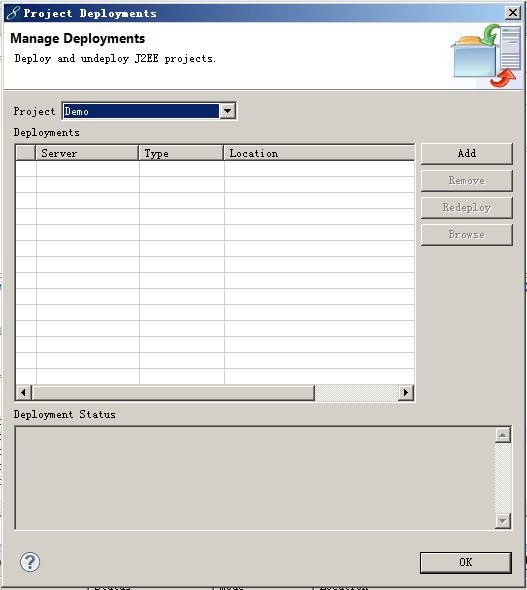


图 11 部署应用 图 12 选择 Web 服务器

框选择要部署的 Web 服务器，通过“Server”列表进行选择。选择后点击图 12 的“Finish” 按钮，并在图 11 点击“OK”按钮，完成部署。

完成部署后，再次打开“Server”选项卡，在图 12 中选择的 Web 服务器下查看是否有部署的工程，如图 13。

1. 运行应用并浏览

在 Server 选项卡中，选择待启动的服务器（Tomcat 7.x）点击右上方的“”按钮启动服务器。然后打开“Console”选项卡如图 14；查看启动日志；若其中没有错误“Error”信息，

并看到“信息: Server startup in xxx ms”的信息，表示服务启动成功。

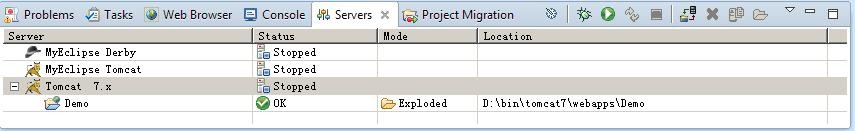


图 13 部署成功的 Server 选项卡

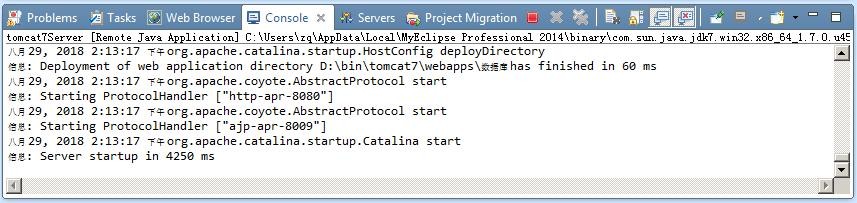


图 14 服务器启动日志

在浏览器地址栏中输入：http://localhost:8080/Demo 即可进行浏览。

3、练习页面跳转及通过 get 方法（URL）进行参数传递。

新建 get.jsp 页面，显示界面如图 15，页面中有 3 个超级连接，点击不同的超级连接在页面 result.jsp 页面中显示用户点击了那个链接，显示结果如图 16。

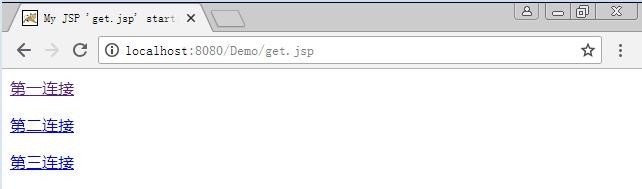


图 15 get.jsp 页面



图 16 result.jsp 页面

***Get.jsp***

*<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>*

*<%*

*String path = request.getContextPath();*

*String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";*

*%>*

*<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">*

*<html>*

*<head>*

*<base href="<%=basePath%>">*

*<title>My JSP 'get.jsp' starting page</title>*

*<meta http-equiv="pragma" content="no-cache">*

*<meta http-equiv="cache-control" content="no-cache">*

*<meta http-equiv="expires" content="0">*

*<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">*

*<meta http-equiv="description" content="第一链接">*

*<!--*

*<link rel="stylesheet" type="text/css" href="styles.css">*

*-->*

*</head>*

*<body>*

*<p align="left"><a href ="result.jsp?name=1">第一链接</a></p> <br>*

*<p align="left"><a href ="result.jsp?name=2">第二链接</a></p> <br>*

*<p align="left"><a href ="result.jsp?name=3">第三链接</a></p> <br>*

*</body>*

*</html>*

**result.jsp**

*<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>*

*<%*

*String path = request.getContextPath();*

*String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";*

*%>*

*<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">*

*<html>*

*<head>*

*<base href="<%=basePath%>">*

*<title>My JSP 'result.jsp' starting page</title>*

*<meta http-equiv="pragma" content="no-cache">*

*<meta http-equiv="cache-control" content="no-cache">*

*<meta http-equiv="expires" content="0">*

*<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">*

*<meta http-equiv="description" content="This is my page">*

*<!--*

*<link rel="stylesheet" type="text/css" href="styles.css">*

*-->*

*</head>*

*<body>*

*<% String sadf = (String)request.getParameter("name");*

*out.print(sadf);*

***int*** *i =Integer.parseInt(sadf);*

***if****(i==1)*

*out.println(("你点击了 \"第一连接\"....") + "<br />");*

***if****(i==2)*

*out.println(("你点击了 \"第二连接\"....") + "<br />");*

***if****(i==3)*

*out.println(("你点击了 \"第三连接\"....") + "<br />");*

*%>*

*</body>*

*</html>*

4、表单及 POST 方法提交参数

创建表单页面 form.jsp，界面如图 17，点击“确定”按钮后将表单内容提交到 do.jsp 处理提交结果，结果页面如图 18。



图 17 表单 form.jsp 页面

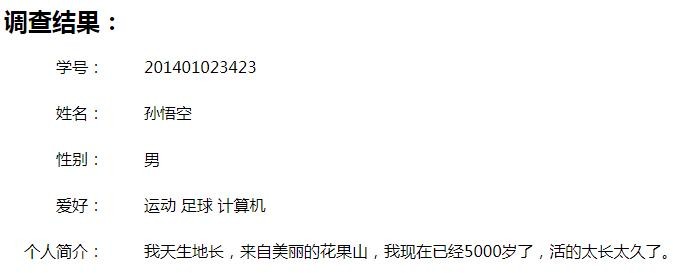


图 18 表单处理结果 do.jsp 页面

**form.jsp**

*<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>*

*<%*

*String path = request.getContextPath();*

*String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";*

*%>*

*<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">*

*<html>*

*<head>*

*<base href="<%=basePath%>">*

*<title>My JSP 'form.jsp' starting page</title>*

*<meta http-equiv="pragma" content="no-cache">*

*<meta http-equiv="cache-control" content="no-cache">*

*<meta http-equiv="expires" content="0">*

*<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">*

*<meta http-equiv="description" content="This is my page">*

*<!--*

*<link rel="stylesheet" type="text/css" href="styles.css">*

*-->*

*</head>*

*<body>*

*<form action="do.jsp" method="post">*

*<fieldset>*

*<legend align="left">个人爱好调查</legend><br>*

*&emsp;&emsp;学号：<input name="studentnu" type="text"/><br><br>*

*&emsp;&emsp;姓名：<input name="uname" type="text"/><br><br>*

*&emsp;&emsp;性别：<input name="sex" value="男" type="radio"/>男*

*<input name="sex" value="女" type="radio"/>女<br><br>*

*&emsp;&emsp;爱好：<input type="checkbox" name="hobs" value="文学">文学*

*<input type="checkbox" name="hobs" value="运动">运动*

*<input type="checkbox" name="hobs" value="旅游">旅游*

*<input type="checkbox" name="hobs" value="足球">足球*

*<input type="checkbox" name="hobs" value="计算机">计算机*

*<input type="checkbox" name="hobs" value="网游">网游<br><br>*

*&emsp;&emsp;个人简介：<br>*

*&emsp;&emsp;&emsp;&emsp;<textarea rows="4" cols="50" name="resume"> </textarea><br>*

*&emsp;&emsp;&emsp;&emsp;&emsp;<input type="submit" value="提交" >*

*</fieldset>*

*</form>*

*</body>*

*</html>*

**do.jsp**

*<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>*

*<%*

*String path = request.getContextPath();*

*String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";*

*%>*

*<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">*

*<html>*

*<head>*

*<base href="<%=basePath%>">*

*<title>My JSP 'do.jsp' starting page</title>*

*<meta http-equiv="pragma" content="no-cache">*

*<meta http-equiv="cache-control" content="no-cache">*

*<meta http-equiv="expires" content="0">*

*<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">*

*<meta http-equiv="description" content="This is my page">*

*<!--*

*<link rel="stylesheet" type="text/css" href="styles.css">*

*-->*

*</head>*

*<body>*

*<%*

*request.setCharacterEncoding("UTF-8");*

*String studentnu=request.getParameter("studentnu");*

*String name=request.getParameter("uname");*

*String sex=request.getParameter("sex");*

*String[] hobs=request.getParameterValues("hobs");*

*String resume=request.getParameter("resume");*

*String hobStr="";*

***for****(String hob:hobs){*

*hobStr+=hob+" ";*

*}*

*%>*

*<fieldset>*

*<h2>调查结果</h2>*

*&emsp;&emsp;学号：<%=studentnu %><br /><br>*

*&emsp;&emsp;姓名：<%=name %><br/><br>*

*&emsp;&emsp;性别：<%=sex %><br/><br>*

*&emsp;&emsp;爱好：<%=hobStr %><br/><br>*

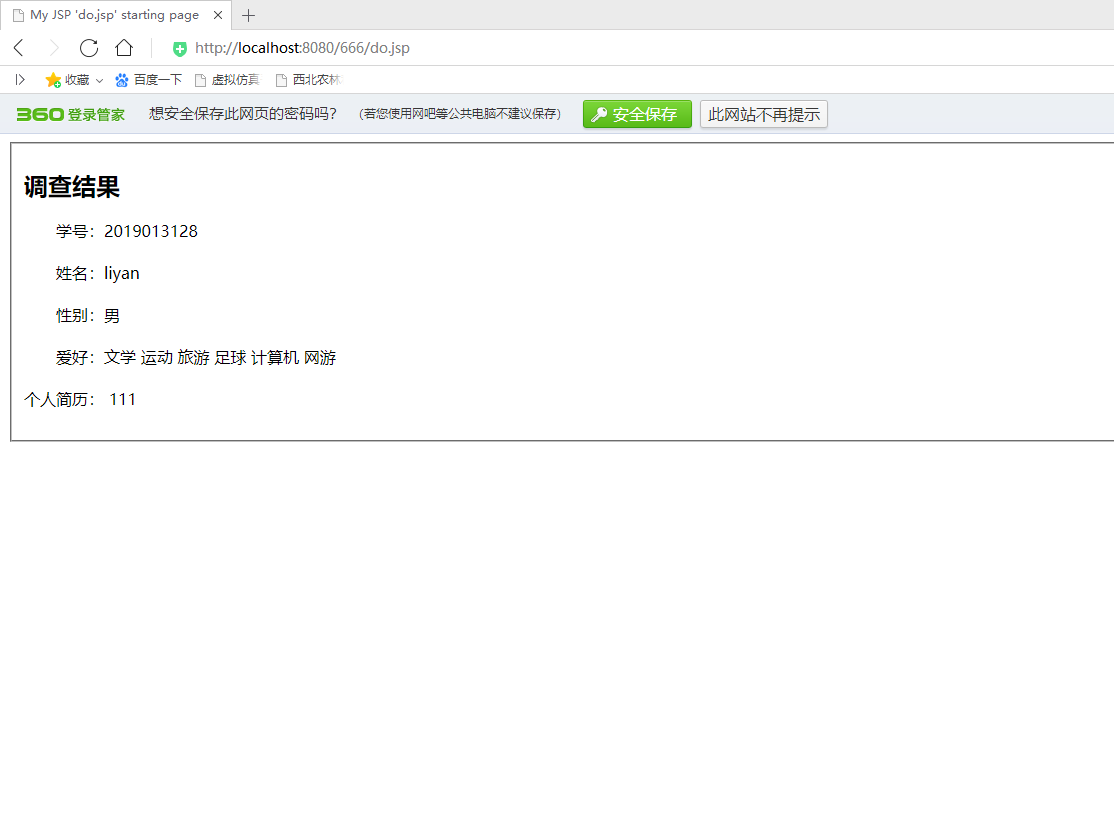
*个人简历：<%=resume %><br/><br>*

*</fieldset>*

*</body>*

*</html>*

**运行结果：**

****

5、登录验证及 session 对象使用

在登录页面（login.jsp）中输入用户名及口令并点击“登录”按钮，将输入的用户名和口令提交到 check.jsp 页面处理验证，若用户名和口令正确则自动跳转到管理页面（admin.jsp），若验证失败则跳转回登录页面（login.jsp）并显示错误信息。若用户在地址栏中直接输入管理页面的 URL 地址进行访问，管理页面要验证用户是否登录，若登录则显示管理页面，否则跳转到登录页面。

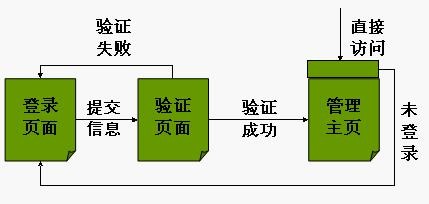


图 19 登录验证流程图

示例代码：

# 登录页面 login.jsp，代码如下：

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "<http://www.w3.org/TR/html4/loose.dtd>">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Insert title here</title>

</head>

<body>

<h2 align="center">系统登录</h2>

<%

//显示登录错误信息

Object errMsg = session.getAttribute("errMsg");

**if**(errMsg != **null**){

%>

<div style="color:red;text-align:center">

<%=errMsg%>

</div>

<%

session.removeAttribute("errMsg");

}

%>

<form method="post" action="check.jsp">

<p align="center">用户名：

<input type="text" name="uName">

</p>

<p align="center">口&nbsp;&nbsp;令：

<input type="password" name="uPassword">

</p>

<p align="center">

<input type="submit" name="submit" value="登录">

</p>

</form>

</body>

</html>

# 验证页面，check.jsp：

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "<http://www.w3.org/TR/html4/loose.dtd>">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Insert title here</title>

</head>

<body>

<%

//确保获取的中文不会乱码request.setCharacterEncoding("UTF-8");

//获取客户端提交的用户名

String uName = request.getParameter("uName");

//获取客户端提交的口令

String uPassword = request.getParameter("uPassword");

**if**("张三".equals(uName) && "123456".equals(uPassword))

{

//将用户名保存到会话中，后面页面通过检查此对象的存在与否判断用户是否登录，并将用户名显示到页面

session.setAttribute("uName",uName);

//重定向到被保护页面response.sendRedirect("admin.jsp");

}

# else{

//设置错误信息，用于登录页面提示session.setAttribute("errMsg","用户名或口令不正确...");

//重定向到登录页面response.sendRedirect("login.jsp");

}

%>

</body>

</html>

# 管理页面：admin.jsp

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "<http://www.w3.org/TR/html4/loose.dtd>">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Insert title here</title>

</head>

<body>

<%

Object uName = session.getAttribute("uName");

//判断用户是否登录

**if**(uName == **null**){

session.setAttribute("errMsg","你还没登录，请登录...");

//重定向到登录页面response.sendRedirect("login.jsp");

}

%>

<h1>嗨！<%=uName%>，欢迎来到天涯海角！！</h1>

</body>

</html>

# 思考题：

在显示错误信息时，代码中使用的时 session 的属性（Attribute）来保存错误信息的，能否使用 request 活 application 的属性进行保存，为什么？

**Request不可以，application可以。**

**因为request的作用范围是一次请求中的，每一个新请求都会重新创建这个范围的对象；**

**Application的作用范围是整个web程序运行期间，所以可以保存。**