**实验二 用户与企业信息添加**

## **实验目标**

本实验任务是实现“Q-ITOffer”锐聘网站后台管理系统的用户添加和企业添加功能：

* 【实验2-1】 用户添加。
* 【实验2-2】 企业添加。

## **【实验2-1】用户添加**

### 知识点回顾

* Servlet的创建
* Servlet处理Form表单请求数据
* Servlet响应的重定向和请求转发
* HttpServletRequest对象获取请求参数的中文问题

### 功能描述

实现“Q-ITOffer”锐聘网站后台管理系统的使用用户的添加功能。

### 实验分析

1. 创建用户添加表单页面userAdd.html，页面运行效果如图2- 3所示。
2. 创建处理用户添加请求的Servlet文件UserAddServlet.java对用户添加表单提交的请求数据进行获取，随后调用JavaBean类User.java对请求数据进行封装，然后将封装好的JavaBean对象传递给数据库操作类UserDAO.java进行数据的保存；数据保存成功后，将本次请求响应到用户列表页面userList.html。
3. 创建用于封装用户信息数据的JavaBean类User.java，将请求的多个用户信息数据封装为一个JavaBean对象方便数据的传递。
4. 创建对用户信息进行数据库添加操作的DAO类UserDAO.java。
5. 上述各资源组件间的关系如图2- 1所示。



图2- 1 用户添加功能组件关系图

### 实验步骤

1. 仿照【实验1-2】中的实验步骤创建Java Web项目“Q\_ITOffer\_Experiment02”，同时将项目所需的模板素材：图片、样式表、js文件分别添加到项目工程目录images、css和js中；在WebContent目录下新建manage子目录，用于存放用户登陆成功后所操作的后台管理页面，这里首先将后台主界面的模板页面：main.html、top.html、left.html、index.html添加到此目录下，然后将用户添加功能的入口操作页面userList.html页面添加到此目录下。

上述操作完成后，启动服务器，运行项目中的main.html页面，在左侧菜单栏中选择“用户管理”子菜单，运行后的userList.html模板页面效果如图2- 2所示。此页面中“添加”按钮即为调用用户添加功能页面的入口。



图2- 2 用户添加操作入口

1. 在项目工程目录manage下，创建用户添加表单页面userAdd.html，功能实现代码如下所示。

**【代码2- 1】userAdd.html**

**<form name="frm" action="../UserAddServlet" method="post">**

<div class="tabson">

<ul class="forminfo">

<li>

<label>登录名称<b>\*</b></label>

**<input name="userLogname" type="text" class="dfinput"**

**style="width:518px;"/>**

</li>

<li>

<label>登录密码<b>\*</b></label>

**<input name="userPwd" type="password" class="dfinput"**

**style="width:518px;"/>**

</li>

<li>

<label>真实姓名<b>\*</b></label>

**<input name="userRealname" type="text" class="dfinput"**

**style="width:518px;"/>**

</li>

<li>

<label>邮箱<b>\*</b></label>

**<input name="userEmail" type="text" class="dfinput"**

**style="width:518px;"/>**

</li>

<li>

<label>用户角色<b>\*</b></label>

<div class="vocation">

**<select name="userRole" class="select3">**

**<option value="3">普通用户</option>**

**<option value="2">企业管理员</option>**

**<option value="1">系统管理员</option>**

**</select>**

</div>

</li>

<li>

<label>用户状态<b>\*</b></label>

**<input name="userState" type="radio" value="1" checked="checked"/>启用**

**<input name="userState" type="radio" value="0"/>禁用**

</li>

<li> <label>&nbsp;</label>

**<input name="" type="submit" class="btn" value="添加"/>**

</li>

</ul>

</div>

</form>

userAdd.html页面运行效果如图2- 3所示。



图2- 3 userAdd.html运行效果

1. 在项目工程目录src🡪com.qst.itofferbacker.servlet下创建处理用户添加请求的Servlet文件UserAddServlet.java，功能实现代码如下所示。

**【代码2- 2】UserAddServlet.java**

package com.qst.itofferbacker.servlet;

import com.qst.itofferbacker.dao.UserDAO;

import com.qst.itofferbacker.javabean.User;

/\*\*

\* 用户信息添加Servlet

\* @公司 青软实训

\* @作者 fengjj

\*/

@WebServlet("/UserAddServlet")

public class UserAddServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

public UserAddServlet() {

super();

}

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// 设置请求字符编码

request.setCharacterEncoding("UTF-8");

// 获取用户添加信息

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

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

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

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

int userRole = (request.getParameter("userRole") == null) ? 3 : Integer.parseInt(request.getParameter("userRole"));

int userState = (request.getParameter("userState") == null) ? 1: Integer.parseInt(request.getParameter("userState"));

// 定义一个用来封装用户信息的JavaBean

User user =

new User(userLogname,userPwd,userRealname,userEmail,userRole,userState);

// 用户信息添加

UserDAO dao = new UserDAO();

dao.save(user);

// 添加成功，重定向到响应页面

response.sendRedirect("manage/userList.html");

}

}

1. 在项目工程目录src🡪com.qst.itofferbacker.javabean目录中创建实现用户信息封装的JavaBean类User.java，具体实现代码如下所示。

**【代码2- 3】User.java**

package com.qst.itofferbacker.javabean;

/\*\*

\* 后台管理用户JavaBean

\* @公司 青软实训

\* @作者 fengjj

\*/

public class User {

private int userId;

private String userLogname;

private String userPwd;

private String userRealname;

private String userEmail;

private int userRole;

private int userState;

public User(){

}

public User(String userLogname, String userPwd,

String userRealname, String userEmail, int userRole, int userState) {

super();

this.userLogname = userLogname;

this.userPwd = userPwd;

this.userRealname = userRealname;

this.userEmail = userEmail;

this.userRole = userRole;

this.userState = userState;

}

public User(int userId, String userLogname, String userPwd,

String userRealname, String userEmail, int userRole, int userState) {

super();

this.userId = userId;

this.userLogname = userLogname;

this.userPwd = userPwd;

this.userRealname = userRealname;

this.userEmail = userEmail;

this.userRole = userRole;

this.userState = userState;

}

......省略setter和getter方法

}

1. 在UserAddServlet.java中通过UserDAO.java实现对用户信息的数据库保存操作，在项目工程目录src🡪com.qst.itofferbacker.dao包下创建此DAO，实现代码如下所示。

**【代码2- 4】userDAO.java**

package com.qst.itofferbacker.dao;

import com.qst.itofferbacker.javabean.User;

import com.qst.itofferbacker.util.DBUtil;

/\*\*

\* 后台管理员用户信息数据库操作类

\* @公司 青软实训

\* @作者 fengjj

\*/

public class UserDAO {

/\*\*

\* 用户添加

\* @param user

\*/

public void save(User user) {

Connection conn = DBUtil.getConnection();

PreparedStatement pstmt = null;

String sql = "INSERT INTO tb\_users(user\_id,user\_logname,user\_pwd,

user\_realname,user\_email,user\_role,user\_state"

+ ") VALUES(SEQ\_ITOFFER\_USERS.NEXTVAL,?,?,?,?,?,?)";

try {

pstmt = conn.prepareStatement(sql);

pstmt.setString(1, user.getUserLogname());

pstmt.setString(2, user.getUserPwd());

pstmt.setString(3, user.getUserRealname());

pstmt.setString(4, user.getUserEmail());

pstmt.setInt(5, user.getUserRole());

pstmt.setInt(6, user.getUserState());

pstmt.executeUpdate();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

## **【实验2-2】企业添加**

### 知识点回顾

* Servlet的创建
* Servlet处理Form表单请求数据
* HttpServletRequest对象获取请求参数的中文问题
* HttpServletResponse对象响应输出中文问题
* Form表单的文件上传
* 注解@MultipartConfig的使用

### 功能描述

实现“Q-ITOffer”锐聘网站后台管理系统的企业添加功能。

### 实验分析

1. 创建企业信息添加表单页面companyAdd.html，页面运行效果如图2- 7所示。
2. 创建处理企业添加请求的Servlet文件CompanyAddServlet.java，对企业信息添加表单提交的请求数据进行获取，由于企业添加功能中包含企业宣传图片的上传功能，因此需要注意实现对上传文件数据流的获取和文件的保存，随后调用JavaBean类Company.java对请求数据进行封装，然后将封装好的JavaBean对象传递给数据库操作类CompanyDAO.java进行数据的保存，数据保存成功后，将本次请求响应到企业列表页面companyList.html。
3. 创建用于封装企业信息数据的JavaBean文件Company.java，将请求的多个企业信息数据封装为一个JavaBean对象方便数据的传递。
4. 创建对企业信息进行数据库添加操作的DAO类CompanyDAO.java。
5. 上述各资源组件间的关系如图2- 4所示。



图2- 4 企业添加功能组件关系图

### 实验步骤

1. 在项目“Q\_ITOffer\_Experiment02”的WebContent🡪manage目录下添加企业列表模板页面companList.html。启动服务器，运行项目中的main.html页面，在左侧菜单栏中选择“企业管理”子菜单，运行后的companyList.html模板页面效果如图2- 5所示，此页面中“添加”按钮即为调用企业添加功能页面的入口。



图2- 5 企业添加入口页面运行效果

1. 在项目工程目录manage下，创建企业信息添加表单页面companyAdd.html，在企业信息添加表单中，对于企业简介这种大段的文本的编辑，使用<textarea>标签并不能方便的实现对文本样式的渲染，因此可以采用第三方提供的文本编辑器插件来完成对文本的编辑，这里我们使用CKEditor插件来实现，其官方下载网址为：http://ckeditor.com/download，下载页面如图2- 6所示。下载完成后，将下载的ckeditor文件夹添加到WebContent目录中，同时在companyAdd.html代码中通过引入此插件的核心js文件ckeditor.js来进行使用。companyAdd.html的功能实现代码如【代码2- 5】所示。

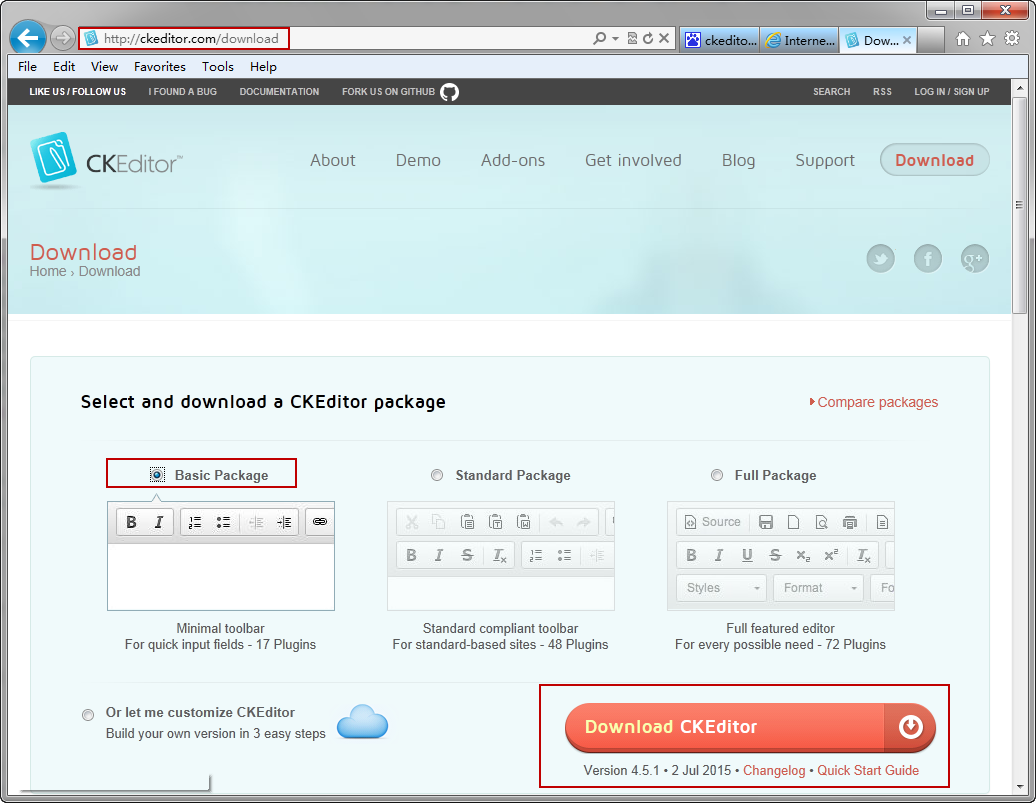


图2- 6 CKEditor下载页面

**【代码2- 5】companyAdd.html**

<!doctype html>

<html>

<head>

<meta charset="utf-8">

<title>添加企业</title>

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

**<script type="text/javascript" src="../ckeditor/ckeditor.js"></script>**

<script type="text/javascript">

**function validate(){**

**if(document.getElementById("companyName").value == ""){**

**alert("企业名称不能为空！");**

**document.getElementById("companyName").focus();**

**return false;**

**}**

**if(document.getElementById("companyArea").value == ""){**

**alert("企业所在地不能为空！");**

**document.getElementById("companyArea").focus();**

**return false;**

**}**

**if(document.getElementById("companySize").value == ""){**

**alert("企业规模不能为空！");**

**document.getElementById("companySize").focus();**

**return false;**

**}**

**if(document.getElementById("companyType").value == ""){**

**alert("企业性质不能为空！");**

**document.getElementById("companyType").focus();**

**return false;**

**}**

**if(document.getElementById("companyPic").value == ""){**

**alert("请选择企业宣传图片！");**

**document.getElementById("companyPic").focus();**

**return false;**

**}**

**return true;**

**}**

</script>

</head>

<body>

......

**<form name="frm" action="../CompanyAddServlet" method="post"**

**enctype="multipart/form-data" onsubmit="return validate();">**

<div class="tabson">

<ul class="forminfo">

<li>

<label>企业名称<b>\*</b></label>

**<input name="companyName" type="text" id="companyName" class="dfinput"**

**style="width:518px;"/>**

</li>

<li>

<label>企业所在地<b>\*</b></label>

**<input name="companyArea" type="text" id="companyArea" class="dfinput"**

**style="width:518px;"/>**

</li>

<li>

<label>企业规模<b>\*</b></label>

**<input name="companySize" type="text" id="companySize" class="dfinput"**

**style="width:518px;"/>**

</li>

<li>

<label>企业性质<b>\*</b></label>

**<input name="companyType" type="text" id="companyType" class="dfinput"**

**style="width:518px;"/>**

</li>

<li>

<p>企业简介&nbsp;(不超过1000字)</p>

**<textarea class="ckeditor" cols="50" id="companyBrief"**

**name="companyBrief" rows="10"></textarea>**

</li>

<li>

<label>企业招聘状态</label>

<div class="vocation">

**<select name="companyState" class="select3">**

**<option value="1">招聘中</option>**

**<option value="2">已暂停</option>**

**<option value="3">已结束</option>**

**</select>**

</div>

</li>

<li>

<label>显示排序<b></b></label>

**<input name="companySort" type="text" class="dfinput"**

**style="width:100px;"/>**

</li>

<li>

<label>宣传图片<b>\*</b></label>

**<input name="companyPic" id="companyPic" type="file" class="dfinput"**

**style="width:300px; margin-right:5px;"/>**

</li>

<li> <label>&nbsp;</label>

**<input name="" type="submit" class="btn" value="添加"/>**

</li>

</ul>

</div>

**</form>**

</div></div></body>

</html>

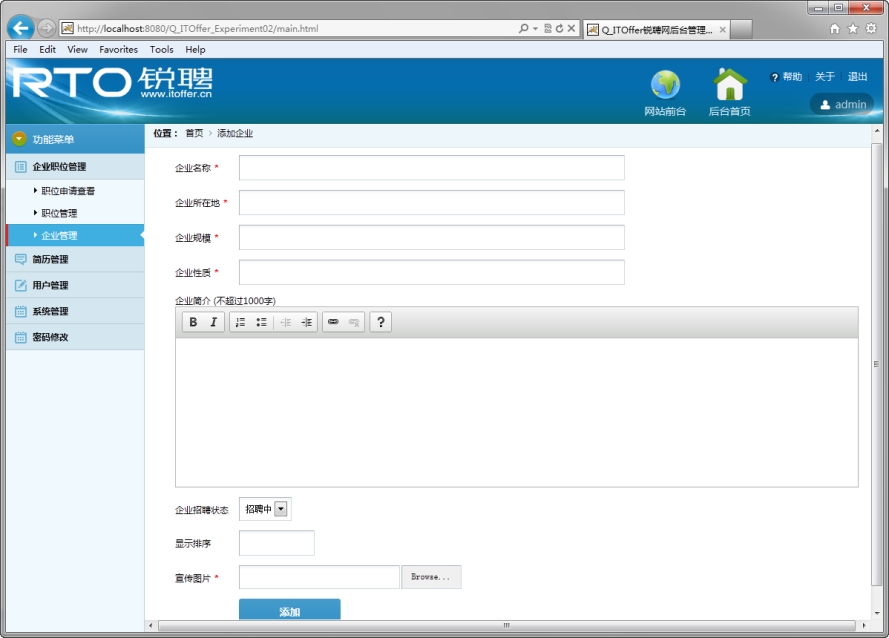


图2- 7 企业添加页面运行效果

1. 在项目工程目录src🡪com.qst.itofferbacker.servlet包下创建处理企业添加请求的Servlet文件CompanyAddServlet.java，其实现代码如下所示。

需要注意的是，为了使前台项目中的企业信息显示可以使用后台上传的企业宣传图片，这里我们可以在Tomcat根目录下创建一个与应用无关的文件上传目录upload，并在此目录下创建用于上传企业宣传图片的companyPic子目录，这样，教材Q\_ITOffer项目中企业宣传图片的查找路径便可更改为此路径，当后台上传新的企业信息及宣传图片时，前台项目便可实时更新。

**【代码2- 6】CompanyAddServlet.java**

package com.qst.itofferbacker.servlet;

import com.qst.itofferbacker.dao.CompanyDAO;

import com.qst.itofferbacker.javabean.Company;

/\*\*

\* 企业信息添加Servlet

\* @公司 青软实训

\* @作者 fengjj

\*/

@WebServlet("/CompanyAddServlet")

@MultipartConfig

public class CompanyAddServlet extends HttpServlet {

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// 设置请求字符编码

request.setCharacterEncoding("UTF-8");

// 设置响应字符编码

response.setContentType("text/html;charset=UTF-8");

// 获取响应字符输出流

PrintWriter out = response.getWriter();

// 获取企业添加信息

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

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

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

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

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

int companyState = (request.getParameter("companyState") == null) ? 1 :

Integer.parseInt(request.getParameter("companyState"));

int companySort = (request.getParameter("companySort") == null) ? 0 :

Integer.parseInt(request.getParameter("companySort"));

// 获取上传文件域

Part part = request.getPart("companyPic");

// 获取上传文件名称

String fileName = part.getSubmittedFileName();

// 为防止上传文件重名，对文件进行重命名

String newFileName = System.currentTimeMillis()

+ fileName.substring(fileName.lastIndexOf("."));

// 将上传的文件保存在服务器根目录下的upload/companyPic子目录下

String filepath = getServletContext().getRealPath("/");

filepath = filepath

.substring(0,filepath.indexOf(getServletContext()

.getServletContextName()));

filepath = filepath + "upload/companyPic";

getServletContext().log("上传路径为：" + filepath);

File f = new File(filepath);

if (!f.exists())

f.mkdirs();

part.write(filepath + "/" + newFileName);

// 定义一个用来封装企业信息的JavaBean

Company company = new Company(companyName,companyArea,companySize,

companyType,companyBrief,companyState,companySort,0,newFileName);

// 企业信息添加

CompanyDAO dao = new CompanyDAO();

dao.save(company);

// 添加成功提示和响应结果导向

out.print("<script type='text/javascript'>");

out.print("alert('企业信息添加成功！');");

out.print("window.location='manage/companyList.html';");

out.print("</script>");

}

}

1. 在项目工程目录src🡪com.qst.itofferbacker.javabean包下创建对企业信息数据进行封装的JavaBean文件Company.java，其实现代码如下所示。

**【代码2- 7】Company.java**

package com.qst.itofferbacker.javabean;

/\*\*

\* 企业信息JavaBean

\* @公司 青软实训

\* @作者 fengjj

\*/

public class Company {

// 企业标识

private int companyId;

// 企业名称

private String companyName;

// 企业所在地区

private String companyArea;

// 企业规模

private String companySize;

// 企业性质

private String companyType;

// 企业简介

private String companyBrief;

// 招聘状态:1招聘中 2已暂停 3已结束

private int companyState;

// 排序序号

private int companySort;

// 浏览数

private int companyViewnum;

// 宣传图片

private String companyPic;

public Company() {

super();

}

public Company(String companyName, String companyArea,

String companySize, String companyType, String companyBrief,

int companyState, int companySort,int companyViewnum,String companyPic) {

super();

this.companyName = companyName;

this.companyArea = companyArea;

this.companySize = companySize;

this.companyType = companyType;

this.companyBrief = companyBrief;

this.companyState = companyState;

this.companySort = companySort;

this.companyViewnum = companyViewnum;

this.companyPic = companyPic;

}

public Company(int companyId, String companyName, String companyArea,

String companySize, String companyType, String companyBrief,

int companyState, int companySort, int companyViewnum,

String companyPic) {

super();

this.companyId = companyId;

this.companyName = companyName;

this.companyArea = companyArea;

this.companySize = companySize;

this.companyType = companyType;

this.companyBrief = companyBrief;

this.companyState = companyState;

this.companySort = companySort;

this.companyViewnum = companyViewnum;

this.companyPic = companyPic;

}

// 省略setter和getter方法

}

1. 在项目工程目录src🡪com.qst.itofferbacker.dao包下创建进行企业信息数据数据库保存的DAO类CompanyDAO.java，其实现代码如下所示。

**【代码2- 8】CompanyDAO.java**

package com.qst.itofferbacker.dao;

import com.qst.itofferbacker.javabean.Company;

import com.qst.itofferbacker.util.DBUtil;

/\*\*

\* 企业信息数据库操作类

\* @公司 青软实训

\* @作者 fengjj

\*/

public class CompanyDAO {

/\*\*

\* 企业信息添加

\* @param company

\*/

public void save(Company company) {

Connection conn = DBUtil.getConnection();

PreparedStatement pstmt = null;

String sql = "INSERT INTO tb\_company("

+ "company\_id,company\_name,company\_area,company\_size,

company\_type,company\_brief,company\_state,company\_sort,

company\_viewnum,company\_pic"

+ ") VALUES(SEQ\_ITOFFER\_COMPANY.NEXTVAL,?,?,?,?,?,?,?,?,?)";

try {

pstmt = conn.prepareStatement(sql);

pstmt.setString(1, company.getCompanyName());

pstmt.setString(2, company.getCompanyArea());

pstmt.setString(3, company.getCompanySize());

pstmt.setString(4, company.getCompanyType());

pstmt.setString(5, company.getCompanyBrief());

pstmt.setInt(6, company.getCompanyState());

pstmt.setInt(7, company.getCompanySort());

pstmt.setInt(8, company.getCompanyViewnum());

pstmt.setString(9, company.getCompanyPic());

pstmt.executeUpdate();

} catch (SQLException e) {

e.printStackTrace();

}

}

}