

# 1.写register.jsp填好表单

<form action=*""* method=*"post"*>

<table>

<tr>

名字：<input type=*"text"* name=*"username"*>

</tr>

<tr>

头像 </p><input type=*"file"* name=*"file"*><br/>

<input type=*"submit"* value=*"注册"*>

</tr>

</table>

</form>

# 2.建立相对的form 和action

# 3.这样我们还是上传不上去，我们的表单有文件这种控件，需要重新制定表单的编码方式，多种编码方式

enctype=*"MULTIPART/FORM-DATA"*

# 4.action 测试

**public** ActionForward execute(ActionMapping mapping, ActionForm form,

HttpServletRequest request, HttpServletResponse response) {

UserForm userForm = (UserForm) form;// **TODO** Auto-generated method stub

String usernameString = userForm.getUsername().toString();

FormFile myphoto = userForm.getMyphoto();

**int** photofilesize = myphoto.getFileSize();

System.*out*.println(myphoto +" ： "+ photofilesize);

//通过formFile 我们可以获得什么信息

**return** **null**;

}

# 5.获取输入流一步到位

**public** **class** RegisterAction **extends** Action{

**public** ActionForward execute(ActionMapping mapping, ActionForm form,

HttpServletRequest request, HttpServletResponse response) {

UserForm userForm = (UserForm) form;// **TODO** Auto-generated method stub

String username = userForm.getUsername();

FormFile myphoto = userForm.getMyphoto();

String photoname = myphoto.getFileName();

**int** photofilesize = myphoto.getFileSize();

System.*out*.println(myphoto +" ： "+ photofilesize);

InputStream is= **null**;

OutputStream os = **null**;

**try**

{

//得到输入流

is = myphoto.getInputStream();

//获得输出流，取得文件的绝对路径

String path = **this**.getServlet().getServletContext().getRealPath("/images");

System.*out*.println(path);

System.*out*.println(path);

System.*out*.println(path);

System.*out*.println(path);

os =**new** FileOutputStream(path+"\\"+photoname);

System.*out*.println(os+"\n\\n\n\n\\n\\\n\\");

**int** len = 0;

**byte**[] buffer = **new** **byte**[1024];

//这里的int代表实际读取了多少个字节

**while** ((len=is.read(buffer))>0) {

os.write(buffer, 0, len);

}

} **catch** (Exception e) {}**finally**{

**try** {

is.close();

os.close();

} **catch** (IOException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

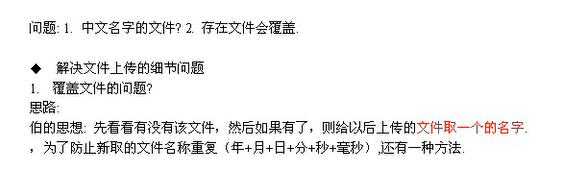
**return** **null**;

}

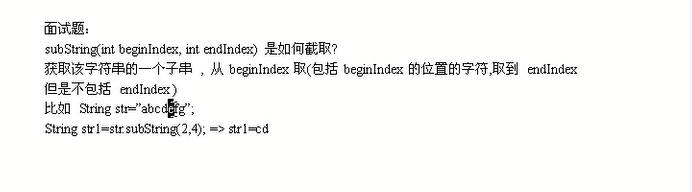
}

# 6.上传细节

## 1.保证文件名不重复



2.



## 1.使用uuid来制作

**public** **class** TestUUID {

**public** **static** String getUUIdName(String filename){

String uuid = UUID.*randomUUID*().toString();

// System.out.println(uuid);

//现在考虑 将文件名filename.jpg 变为 uuId.jpg

// String filename= "abcdefghijklmn.jpg";

**int** begainName = filename.lastIndexOf(".");

//取得文件的前面的名字

// String newName = filename.substring(0, filename.lastIndexOf("."));

// 获取后缀名 因为是包头不包尾所以就直接length

// String newName = filename.substring(begainName, filename.length());

//制作新名字

String newName = uuid+filename.substring(begainName, filename.length());

System.*out*.println(newName);

**return** newName;

}

}

## 2.如果文件名是中文也会出错，所以我们想到了使用过滤器，发现成功了

# 7.下载

## 1.下载的文件名将会复原

## 2.所以讲会将我们的数据保存到数据库中

create table users(

username varchar(20) unique not null

,photoBefore varchar(120) not null,

photoNow varchar(128) not null);

## 3.然后我们建立一个domain和一个userServiece

**public** **class** UserService {

**public** **boolean** addUser(Users user){

**boolean** flag = **true**;

**try** {

String sql = "insert into users values(?,?,?)";

String []parameter= {user.getUsername(),user.getPhotoBefore(),user.getPhotoNow()};

SqlHelper.*executeUpdate*(sql, parameter);

} **catch** (Exception e) {

// **TODO**: handle exception

e.printStackTrace();

**return** **false**;

}

**return** **true**;

}

}

## 4.到了action中进行添加数据库

**while** ((len=is.read(buffer))>0) {

os.write(buffer, 0, len);

}

UserService us = **new** UserService();

Users users = **new** Users();

users.setUsername(username);

users.setPhotoBefore(photoname);

users.setPhotoNow(photonameNew);

**if**(us.addUser(users)==**true**){

**return** mapping.findForward("ok");

}

## 5.到了config中添加一个图片上传失败的界面

**if**(us.addUser(users)==**true**){

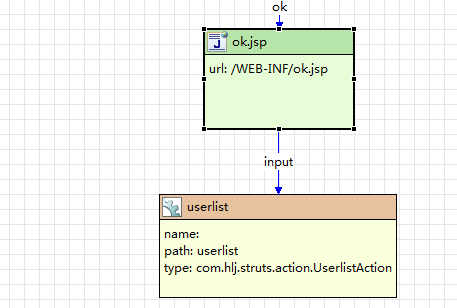
**return** mapping.findForward("ok");

}**else**{

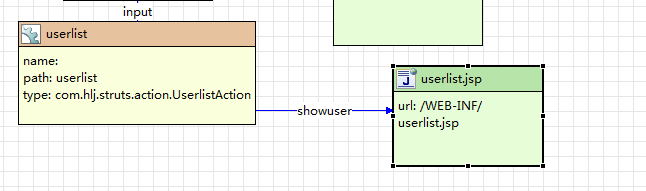
**return** mapping.findForward("err");

}

# 8.建立一个action UserList,不需要表单



# 9。UserList.jsp



## 让ok界面跳转到这个action中

<h1>register sussesful</h1>

<a href=*"/StrutsScXz/userlist.do"*>点击查看用户列表</a>

# 10,User service来获得内容了

**public** ArrayList getUsers(){

System.*out*.println("正在使用 getUsers ");

SqlHelper sqlHelper = **new** SqlHelper();

String sql = "select \* from users";

ArrayList arrayList =sqlHelper.executeQuery(sql,**null**);

ArrayList<Users> users = **new** ArrayList<Users>();

**if**(arrayList.size()!=0){

**for** (**int** i = 0 ; i< arrayList.size() ; i++){

Object objects[] = (Object[])arrayList.get(i);

Users user = **new** Users();

user.setUsername(objects[0].toString());

user.setPhotoBefore(objects[1].toString());

user.setPhotoNow(objects[2].toString());

users.add(user);

}

}

**return** users;

}

# 11.到了action中进行调用

**public** ActionForward execute(ActionMapping mapping, ActionForm form,

HttpServletRequest request, HttpServletResponse response) {

// **TODO** Auto-generated method stub

//开始准配数据

System.*out*.println("已经进入UserlistAction了 这个界面");

UserService userService = **new** UserService();

ArrayList<Users> users = userService.getUsers();

System.*out*.println(users.size());

request.setAttribute("users", users);

**return** mapping.findForward("showuser");

}

# 12.开始显示

<h1>用户列表</h1>

<c:forEach items=*"*${users}*"* var=*"u"*>

${u.username} <img alt=*""* src=*"/StrutsScXz/images/*${u.photoNow}*"*><a href=*"/StrutsScXz/downfile.do?username=*${u.username}*"*>下载</a>

<br/>

</c:forEach>

# 13. username得到我们的user的主要内容（userserviece）

**public** Users getUser(String username){

String sql = "select \* from users where username = ?";

String parameter[] ={username};

ArrayList arrayList =SqlHelper1.*executeQueryArrayList*(sql, parameter);

System.*out*.println("这个的大小为"+arrayList.size());

Users user = **new** Users();

**if**(arrayList.size()!=0){

**for** (**int** i = 0 ; i< arrayList.size() ; i++){

Object objects[] = (Object[])arrayList.get(i);

user.setUsername(objects[0].toString());

user.setPhotoBefore(objects[1].toString());

user.setPhotoNow(objects[2].toString());

}

}

**return** user;

}

# 14.制作下载，还是要建立一个action

**public** ActionForward execute(ActionMapping mapping, ActionForm form,

HttpServletRequest request, HttpServletResponse response) {

// **TODO** Auto-generated method stub

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

System.*out*.println("传过来的参数为 username 为" +username);

UserService userService = **new** UserService();

Users user = userService.getUser(username);

System.*out*.println(user.getUsername());

//开始下载了

OutputStream oStream = **null**;

FileInputStream fis = **null**;

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

response.setHeader("Content-Disposition", "attachment;filename="+user.getPhotoBefore());

//打开文件，说明下web下载文件的原理

//1.获取要下载文件的全路径

String path1 = getServlet().getServletContext().getRealPath("/images/"+user.getPhotoNow());

//这个时下载D盘的图片

//File file = new File("D://a.jpg");

//String path2 = file.getPath();

//System.out.println(path2);

System.*out*.println(path1);

//下面就是打开这个文件(首先要读出了)

**try** {

fis = **new** FileInputStream(path1);

oStream = response.getOutputStream();

// 通过文件输入流写入文件

**byte** buffer[] = **new** **byte**[1024];

**int** len =0; //表示实际每次读取了多少个字节

**while**((len=fis.read(buffer))>0){

oStream.write(buffer,0,len);

}

} **catch** (FileNotFoundException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

} **catch** (IOException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**try** {

fis.close();

oStream.close();

} **catch** (IOException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**return** **null**;

}

# 15. ava.net.URLEncoder.*encod*图片的名字是中文的话，下载下来将会是乱码

response.setHeader("Content-Disposition", "attachment;filename="+java.net.URLEncoder.*encode*(user.getPhotoBefore(),"utf-8"));