

Struts2 框架基础

L5 《Struts2 补充》上机手册

1. 内容回顾（3-5 分钟）

1. 理解 Struts2 工作原理
2. Struts2 文件上传/下载

2. 上机目标

1. 掌握 Struts2 文件上传实现
2. 掌握 Struts2 文件下载实现

3. 上机任务

1. 实现文件上传/下载（55 分钟）

功能要求:实现上传文件到服务器。上传成功后显示回执信息包括文件的名称，实际大小，类型等，并且提供下载的方式下载刚上传的文件到本地。

参考实现步骤:

(1) 编写文件上传表单页面 `upload.jsp`

```
<s:form action="uploadfile" enctype="multipart/form-data" method="post">
  <s:textfield name="uname" label="用户名"/>
  <s:file name="upload" label="上传文件"></s:file>
  <s:submit value="确定"></s:submit>
</s:form>
```

说明:

文件上传表单的 `enctype` 属性必须设置成 `multipart/form-data`
表单提交方式必须是 `POST`
这里使用 `<s:file>` 标签，参数字段名 `upload`

(2) 编写处理文件上传的 Action 组件

Struts2 框架基础

首先，增加与上传字段相关的三个属性

```
private File upload;  
private String uploadFileName;  
private String uploadContentType;
```

说明：这三个属性必须和上传文件字段名保持一致。其中黄色标记的必须和上传字段名一样。其中 upload 属性可读取上传文件数据内容

uploadFileName 可获取上传文件名

uploadContentType 可获取上传文件内容类型

再者，在 Action 的处理上传的业务方法中将临时文件中的数据保存到指定路径的文件中

```
FileInputStream fis = new FileInputStream(upload);  
String destfile = request.getSession().getServletContext().getRealPath("/") +  
    "upload/" + uploadFileName;  
FileOutputStream fos = new FileOutputStream(destfile);  
byte[] buffer = new byte[1024];  
int len = 0;  
while((len=fis.read(buffer))!=-1){  
    fos.write(buffer, 0, len);  
}  
fos.close();  
fis.close();
```

说明：

`request.getSession().getServletContext().getRealPath("/")` 目的是获取服务器所在的磁盘绝对路径，以便将上传文件保存到服务器所在的目录下，这里是服务器应用所在的 upload 目录下。

(3) 编写上传成功回执页面 uploadresult.jsp

```
<h2>文件上传</h2>  
用户名:<s:property value="uname"/><br/>  
上传文件名:<s:property value="uploadFileName"/><br/>  
类型:<s:property value="uploadContentType"/>
```

Struts2 框架基础

(4) 配置 Action 组件

```
<!-- 文件上传下载 -->  
<action name="uploadfile" class="demo.web.UploadAction">  
    <result>/uploadresult.jsp</result>  
</action>
```

(5) 测试

文件上传

用户名:

上传文件:

上传成功，显示回执

文件上传

用户名:QQ
上传文件名:LoginAction_zh_CN.txt
类型:text/plain

到这一步上传的基本功能实现，但回执页面中没有显示文件的实际大小，也没有提供下载的方式。

那么继续...

(6) 实现回执中显示文件实际大小

在 Action 组件中增加一个属性 `fileSize`，表示文件的实际大小，在处理上传的业务方法中获取文件的实际大小信息。

参考代码如下：

Struts2 框架基础

```
long flen = upload.length();
if(flen<1024){
    fileSize = flen + "字节";
}else if(flen>1024 && flen < 1024*1024){
    fileSize = ((long)(flen/1024.0d)*100)/100.0 + "K字节";
}else{
    fileSize = ((long)(flen/(1024.0d*1024)*100))/100.0 + "M字节";
}
```

在回执页面显示文件实际大小:

```
<h2>文件上传</h2>
用户名:<s:property value="uname"/><br/>
上传文件名:<s:property value="uploadFileName"/>
    [<s:property value="fileSize"/>]<br/>
类型:<s:property value="uploadContentType"/>
```

测试结果:

文件上传

用户名:QQ
上传文件名:LoginAction_zh_CN.txt [186字节]
类型:text/plain

(7) 提供文件下载功能

这里有两种实现方式:

一种方式: 直接提供文件超链接, 客户端用户右键选择“另存为...”来下载, 在回执页面添加如下代码即可实现下载功能。

```
<s:a href="/sshdemo/upload/{uploadFileName}">普通下载[选右键另存]</s:a>
```

Struts2 框架基础

另一种方式：使用流方式进行下载。实现方式如下，

首先需要在 Action 组件中增加两个属性：

```
private InputStream inputStream; // 获取下载文件输入流
private String downFileName; // 下载文件名
```

在 Action 中提供处理文件下载的业务方法 `download()`：

```
public String download() throws Exception {
    String dfile = request.getSession().getServletContext().getRealPath("/") +
        "upload/" + downFileName;
    inputStream = new FileInputStream(dfile);
    return "download";
}
```

然后配置 Action 组件，这里不再单独配置，而是采用 Struts2 的简化配置即每个 Action 组件可只配置一次来响应多个不同的业务请求。

```
<!-- 文件上传下载 -->
<action name="uploadfile" class="demo.web.UploadAction">
    <result>/uploadresult.jsp</result>
    <result name="download" type="stream">
        <param name="contentType">application/msword</param>
        <param name="inputName">inputStream</param>
        <param name="contentDisposition">filename="${downFileName}"</param>
    </result>
</action>
```

注意：

`<result name="download" type="stream">` 表示 download 结果视图类型为流 stream。

这里的 UploadAction 组件可处理两种不同业务请求，一个上传一个下载。不能再配置 method 属性。那么 UploadAction 组件如何区分是哪种请求呢？请继续往下看。。。

在回执页面增加一个请求文件下载的超链接：

```
<s:a href="/sshdemo/uploadfile!download?downFileName=%{uploadFileName}">下载</s:a>
```

说明：uploadfile 是 Action 组件的映射名

!download 表示!后面是请求的业务方法名

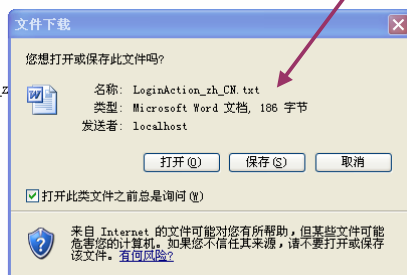
?downFileName= 表示 URL 重写参数

测试效果如下：

文件上传

用户名:QQ
上传文件名>LoginAction_z
类型:text/plain

[普通下载\(选右键另存\)](#)
[下载](#)



Struts2 框架基础

2. 用户管理模块中增加用户图片（25 分钟）

功能要求：在新增用户同时可上传用户的头像。

提示：可参考第 1 节提供的解决方式来实现。

3. 员工信息管理模块中增加新增员工时上传员工照片的功能。

课外完成。