Struts2 文件上传

Struts2 框架为依据“基于表单的 HTML 文件上传”所进行的文件处理上传提供了内置支持。当文件上传时，它通常会存储在临时目录中，然后 Action 类应对其进行处理或移动到固定目录中，以确保数据不会丢失。

注意：服务器可能有适当的安全策略，禁止你写入临时目录以外的目录以及属于 Web 应用程序的目录。

通过一个名为 **FileUpload** 的预定义拦截器可以在 Struts 中上传文件，该拦截器可通过 org.apache.struts2.interceptor.FileUploadInterceptor 类获得，并作为 **defaultStack** 的一部分包含在内。你也将在接下来的内容中看到如何使用它在 struts.xml 文件中设置各种参数。

创建视图文件

创建视图时需要浏览和上传选定的文件。因此，让我们先使用 HTML 上传表单，创建一个允许用户上传文件的 **index.jsp**：

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="s" uri="/struts-tags"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>File Upload</title>
</head>
```

```
<body>
   <form action="upload" method="post" enctype="multipart/form-data">
      <label for="myFile">Upload your file</label>
      <input type="file" name="myFile" />
      <input type="submit" value="Upload"/>
   </form>
</body>
</html>
```

在上面的例子中有几点值得注意。首先，表单的 enctype 设置为 **multipart/form-data**，要使得文件上传拦截器成功处理文件上传，这个就必须设置。然后要注意的是表单的 action 方法上传和文件上传字段的名称（即 **myFile**）。我们需要这些信息来创建 action 方法和 struts 配置。

接下来让我们创建一个简单的 jsp 文件 **success.jsp** 来显示我们文件上传成功后的结果。

```
<%@ page contentType="text/html; charset=UTF-8" %>
<%@ taglib prefix="s" uri="/struts-tags" %>
<html>
<head>
<title>File Upload Success</title>
</head>
<body>
You have successfully uploaded <s:property value="myFileFileName"/>
</body>
</html>
```

以下是结果文件 **error.jsp**，一旦上传文件出错时会使用：

```
<%@ page contentType="text/html; charset=UTF-8" %>
<%@ taglib prefix="s" uri="/struts-tags" %>
<html>
<head>
<title>File Upload Error</title>
</head>
<body>
There has been an error in uploading the file.
</body>
</html>
```

创建 Action 类

接下来，让我们创建一个名为 **uploadFile.java** 的 Java 类，它将负责上传文件并将文件存储在安全的位置：

```
package cn.w3cschool.struts2;

import java.io.File;
import org.apache.commons.io.FileUtils;
import java.io.IOException;

import com.opensymphony.xwork2.ActionSupport;

public class uploadFile extends ActionSupport{
```

```
private File myFile;
private String myFileContentType;
private String myFileFileName;
private String destPath;

public String execute()
{
   /* Copy file to a safe location */
   destPath = "C:/apache-tomcat-6.0.33/work/";

   try{
       System.out.println("Src File name: " + myFile);
       System.out.println("Dst File name: " + myFileFileName);

       File destFile  = new File(destPath, myFileFileName);
       FileUtils.copyFile(myFile, destFile);

   }catch(IOException e){
      e.printStackTrace();
      return ERROR;
   }

   return SUCCESS;
}
public File getMyFile() {
   return myFile;
```

```
   }
   public void setMyFile(File myFile) {
      this.myFile = myFile;
   }
   public String getMyFileContentType() {
      return myFileContentType;
   }
   public void setMyFileContentType(String myFileContentType) {
      this.myFileContentType = myFileContentType;
   }
   public String getMyFileFileName() {
      return myFileFileName;
   }
   public void setMyFileFileName(String myFileFileName) {
      this.myFileFileName = myFileFileName;
   }
}
```

uploadFile.java 是一个非常简单的类。要注意的重点是，FileUpload 拦截器和 Parameters 拦截器为我们承担了所有的重工作量。默认情况下，FileUpload 拦截器为你提供三个参数，它们分别按以下方式命名：

- **[文件名参数]** - 这是用户已上传的实际文件。在这个例子中它将是“myFile”
- **[文件名参数]ContentType** - 这是上传的文件的内容类型。在这个例子中，它将是“myFileContentType”

- **[文件名参数]FileName** - 这是上传的文件的名称。在这个例子中，它将是

  “myFileFileName”

得益于 Struts 拦截器这三个参数均可供我们使用。我们要做的是在 Action 类中创建三个带有正确名称的参数，并使这些变量可以自动连接。所以，在上面的例子中，我们有三个参数和一个 action 方法。如果一切正常，则返回“success” ，否则返回“error” 。

配置文件

以下是控制文件上传过程的 Struts2 配置属性：

序号	属性和说明
1	struts.multipart.maxSize 可接受的上传文件的最大值（以字节为单位），默认值为 250M。
2	struts.multipart.parser 用于上传多部分表单的库，默认为 jakarta。
3	struts.multipart.saveDir 存储临时文件的位置，默认是 javax.servlet.context.tempdir。

你可以在应用程序的 struts.xml 文件中使用 **constant** 标签更改任意一个这些设置，我们可以看以下 **struts.xml** 的示例：

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE struts PUBLIC

"-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"

"http://struts.apache.org/dtds/struts-2.0.dtd">



<struts>

   <constant name="struts.devMode" value="true" />

   <constant name="struts.multipart.maxSize" value="1000000" />
```

```
   <package name="helloworld" extends="struts-default">
   <action name="upload" class="cn.w3cschool.struts2.uploadFile">
       <result name="success">/success.jsp</result>
       <result name="error">/error.jsp</result>
   </action>
   </package>
</struts>
```

因为 **FileUpload** 拦截器是 defaultStack 拦截器的一部分，我们不需要准确的配置它，但你可以在<action>中添加<interceptor-ref>标签。fileUpload 拦截器有两个参数：**maximumSize** 和 **allowedTypes**。 maximumSize 参数是设置所允许的文件大小的最大值（默认约为 2MB）。allowedTypes 参数是所允许的内容（MIME）类型的用逗号分隔的列表，如下所示：

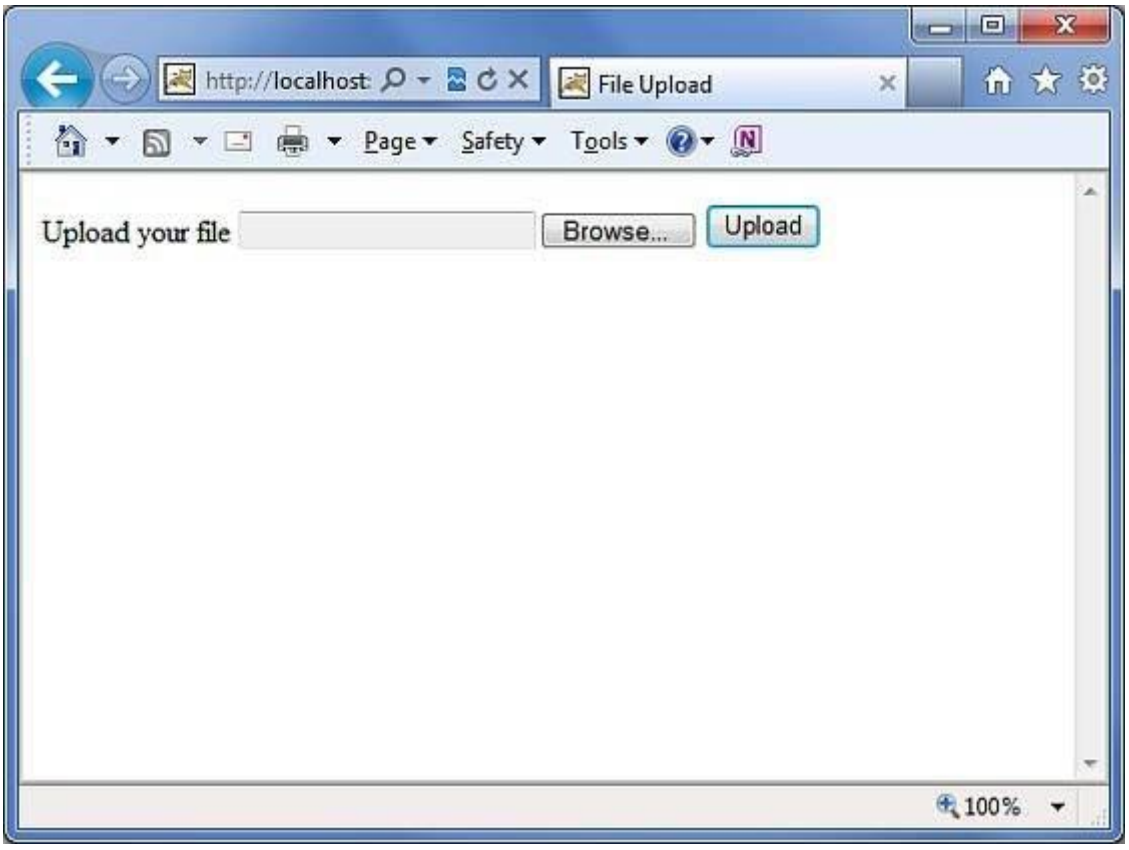
```
   <action name="upload" class="cn.w3cschool.struts2.uploadFile">
       <interceptor-ref name="basicStack">
       <interceptor-ref name="fileUpload">
           <param name="allowedTypes">image/jpeg,image/gif</param>
       </interceptor-ref>
       <result name="success">/success.jsp</result>
       <result name="error">/error.jsp</result>
   </action>
```

以下是 **web.xml** 文件的内容：

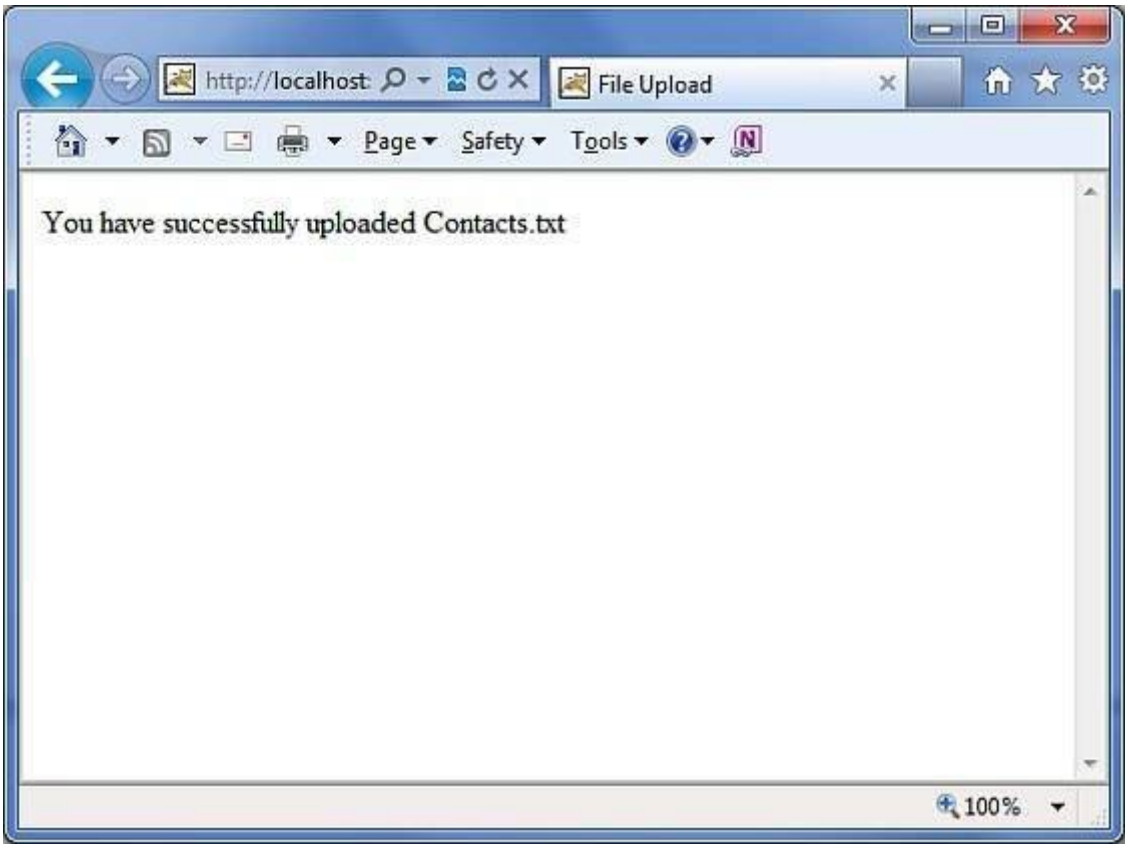
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
   xmlns="http://java.sun.com/xml/ns/javaee"
   xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
   xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
   http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
   id="WebApp_ID" version="3.0">

   <display-name>Struts 2</display-name>
   <welcome-file-list>
      <welcome-file>index.jsp</welcome-file>
   </welcome-file-list>
   <filter>
      <filter-name>struts2</filter-name>
      <filter-class>
         org.apache.struts2.dispatcher.FilterDispatcher
      </filter-class>
   </filter>

   <filter-mapping>
      <filter-name>struts2</filter-name>
      <url-pattern>/*</url-pattern>
   </filter-mapping>
</web-app>
```

现在右键单击项目名称，然后单击**“Export”>“WAR File”**创建 WAR 文件。然后在 Tomcat 的 webapps 目录中部署 WAR 文件。最后，启动 Tomcat 服务器并尝试访问 URL http://localhost:8080/HelloWorldStruts2/upload.jsp，将显示如下界面：



现在使用浏览按钮选择一个文件“Contacts.txt”，然后点击上传按钮，上传文件到服务器，你将看到如下页面。你上传的文件应该保存在 C:\apache-tomcat-6.0.33\work 下。



注意：FileUpload 拦截器会自动删除上传的文件，因此你必须在上传的文件被删除之前将其以编程方式保存在某个位置。

错误信息

fileUplaod 拦截器使用几个默认的错误信息 key：

序号	错误信息 key 和说明
1	struts.messages.error.uploading 无法上传文件时发生的常规错误。
2	struts.messages.error.file.too.large 当上传的文件过大（由 maximumSize 指定）时发生。
3	struts.messages.error.content.type.not.allowed 当上传的文件与指定的预期内容类型不匹配时发生。

你可以在 **WebContent/WEB-INF/classes/messages.properties** 资源文件中覆盖这些消息文本。