文件的上传下载

## 原理

### 文件上传的原理

在TCP/IP中，最早出现的文件上传机制是FTP，它是将文件由客户端发送到服务器端的标准机制。

但是在jsp编程中不能使用FTP方式来上传文件，这是由jsp运行机制所决定的。

通过为表单元素设置method=”post” enctype=”multipart/form-data”属性，让表单提交的数据以二进制编码的方式提交，在接收此请求的Servlet中用二进制流来获取内容，就可以取得上传文件的内容，从而实现文件的上传。

### 表单ENCTYPE属性

Application/x-www-form-urlencoded:这是默认的编码方式，它只处理表单域里的value属性值，采用这种编码方式的表单会将表单域的值处理成URL编码方式。

Multipart/form-data:这种编码方式的表半日会以二进制流的方式来处理表单数据，这种编码方式会把文件域指定文件的内容也封装到请求参数里。

Text/plain:这种方式主要适用于直接通过表单发送邮件的方式。

### 文件下载的原理

STEP1

需要通过HttpServletResponse.setContentType方式设置Content-Type头字段的值，为浏览器无法使用某种方式或激活某个程序来处理的MIME类型，例如，“application/octet-stream”或“application/x-msdownload”等。

SETP2

需要通过HttpServletResponse.setHeader方式设置Content-Dispositon头的值的“attachment;filename=文件名”

STEP3

读取下载文件，调用HttpServletResponse.getOutputStream方法返回的ServletOutputStream对象来向客户端写入附件文件内容。

## 上传实现思路

### 环境搭建

配置web.xml，创建UploadServlet

### 前台页面配置

Form的method设置为post，Enctype配置为multipart/form-data

前台页面

<form action=”” method=”post” enctype=”multipart-data”>

<input type=”file” />

<input type=”submit” value=”提交” />

</form>

### 后台Servlet

#### 文件上传后台实现步骤

STEP1

获取request当中的流信息，保存到临时文件

STEP2

从临时文件当中得到上传的文件名及文件内容起止位置

STEP3

根据文件起止位置，读取上传文件内容，保存到本地。

#### 实现代码

/\*\*

\*首先文件开始上传，读取输入流，保存到临时文件中，这其中包含文件的上传信息和文件内容，需要获取文件内容的开始位置和结束位置，然后将文件内容最终放入需要保存的文件中。

\*/

// 从request当中获取流信息

InputStream inpusStream = request.getInputStream();

// 设置本地临时文件(存入的不仅仅是上传的文件内容，还包括一些上传信息，文件内容需要截取出来)

File tempFile = new File(“本地临时文件路径”);

// 设置指向本地临时文件的输出流

FileOutputStream fos = new FileOutputStream(tempFile);

byte[] buffer = new byte[1024];

int n;

while(( n = inpusStream.read(buffer)) != -1)

{

fos.write(buffer,0,n);

}

// 关闭输出流

fos.close();

// 关闭输入流

inputStream.close();

// 获取上传文件的名称

RandomAccessFile randomAccessFile = new RandomAccessFile(tempFile,”r”);

// 文件第一行信息基本无用

randomAccessFile.readLine();

// 文件第二行信息

String str = randomAccessFile.readLine();

// 获取文件名开始位置

Int beginIndex = str.lastIndexOf(“\\”) + 1;

// 获取文件名结束位置

Int endIndex = str.lastIndexOf(“\””);

// 获取文件名

String filename = str.substring(beginIndex,endIndex);

// 重新定位文件指针到文件头

randomAccessFile.seek(0);

//

Long startPosition = 0 ;

//

Int I = 1;

// 获取文件内容开始位置

While((n = randomAccessFile.readByte()) != -1 && I <= 4)

{

If(n == ‘\n’)

{

startPosition = randomAccessFile.getFilePointer();

I ++;

}

}

startPosition = startPosition – 1;

// 获取文件内容结束位置

randomAccessFile.seek(randomAccessFile.length());

long endPositon = randomAccessFile.getFilePointer();

int j = 1;

while(endPosition >= 0 && j<=2)

{

endPosition --;

randomAccessFile.seek(endPosition);

if(randomAccessFile.readByte() == ‘\n’)

{

j++;

}

}

endPositon = endPosition -1;

// 设置保存上传文件的路径

String realPath = getServletContest.getRealPath(“/”) + “images”;

File fileupload = new File(readPaht);

If(!fileupload.exists())

{

Fileupload.mkdir();

}

File saveFile = new File(realPath,filename);

RandomAccessFile randomFile = new RandomAccessFile(saveFile,”rw”);

// 从临时文件当中读取文件内容（根据起止位置获取）

randomFile.seek(startPosition);

while(startPosition < endPosition)

{

randomFile.write(randomAccessFile.readByte());

startPosition = randomAccessFile.getFilePointer();

}

// 关闭输出流

randomFile.close();

// 关闭输入流

randomAccessFile.close();

// 删除临时文件

tempFile.delete();

## 下载实现思路

### 前台页面开发

通过超链接的方式发起文件下载请求

给后台传递需要下载的文件相应参数

### 配置web.xml

配置web.xml，创建DownloadServlet

### 后台Servlet

设置响应类型及响应头，输出流写入文件内容

#### 实现代码

String realPath = getServletContext().getRealPath(“/”) + “images/”;

// 前台传入的需要下载的文件名

String filename = request.getParameter(“filename”);

File file = new File(realPath + filename);

if(file.exist())

{

// 设置响应类型

response.setContentType(“application/x-msdownload”);

// 设置头信息

Response.setHeader(“Content-Disposition”,”attachment;filename=\”” + filename +”\””);

//

InputStream inpusStream = new FileInputStream(file);

ServletOutputStream servletOutputStream = response.getOutputStream();

byte[] buffer = new byte[1024];

int n;

while((n = inputStream.read(buffer)) != -1)

{

servletOutputStream.write(buffer,0,n);

}

servletOutputStream.close();

inputStream.close();

}

else

{

logger.warn(“文件不存在”)；

}

## 使用SmartUpload组件

### SmarUpload介绍

SmarUpload是由[www.jspsmart.com](http://www.jspsmart.com)网站开发的一套上传组件包，可以轻松的实现文件的上传及下载功能。

使用该组件可以轻松的实现上传文件类型的限制，也可以轻易的取得文件上传的名称、后缀、大小等。

SmartUpload本身也是一个系统提供的jar包。

直接将此包放classpath下即可，也可以拷贝到TOMCAT\_HOME\lib目录中。

## 基于Struts2实现文件上传下载

### 介绍

Struts2默认使用的是Jakara和Common-FileUpload的文件上传框架，因此，如果需要使用Struts2的文件上传功能，由需要在Web应用导入相关jar包

Struts.xml配置拦截器，设置允许上传类型、文件大小等信息。