过滤器 /\*

过滤器链，注册顺序相关

FilterConfig 读取初始化参数

生命周期 随着tomcat启动时，创建过滤器对象

dispatcher 配置 理解

监听器

针对 ServletContext、HttpSession、HttpServletRequest

生命周期相关的

属性相关的

# 一 文件上传 掌握

## 1 准备工作

1）表单的提交方式为post

2）表单的enctype=”multipart/form-data”

3）input的type=”file”

注：enctype 属性表示在请求发送到服务器之前应该如何对表单数据进行编码。

默认值为 "application/x-www-form-urlencoded"。

值描述：

1）application/x-www-form-urlencoded：在发送到服务器之前，所有字符都会进行编码（空格转换为 "+" 加号，特殊符号转换为 ASCII HEX 值）。

2）multipart/form-data：不对字符编码。在使用包含文件上传控件的表单时，必须使用该值

3）text/plain：空格转换为 "+" 加号，但不对特殊字符编码。

## 2 通过request.getInputStream获取上传数据

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

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

//无法获取普通表单控件的数据

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

System.***out***.println(name);//null

ServletInputStream inputStream = request.getInputStream();

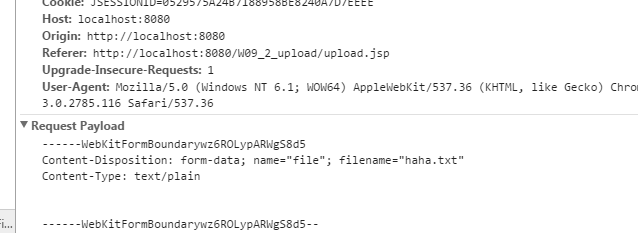
**byte**[] buff = **new** **byte**[1024];

**int** len = inputStream.read(buff);

String str = **new** String(buff, 0, len);

System.***out***.println(str);

}



输出内容：

------WebKitFormBoundarywz6ROLypARWgS8d5

Content-Disposition: form-data; name="file"; filename="haha.txt"

Content-Type: text/plain

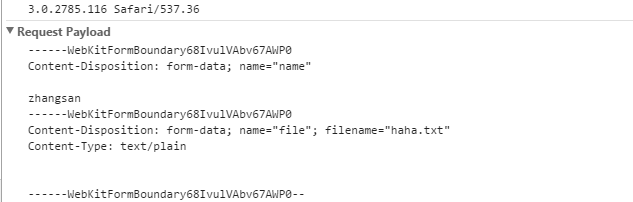
asfasjflsadfj

asdfasjflasjdf

asfjasljfasdlkf

safjaslfdjlsfj

------WebKitFormBoundarywz6ROLypARWgS8d5--



## 3通过fileupload组件进行上传

首先需要导入相关jar文件：

commons-fileupload-1.2.1.jar

commons-io-1.4.jar

核心类：

DiskFileItemFactory：是创建 FileItem 对象的工厂

这个工厂类常用方法：

public DiskFileItemFactory(int sizeThreshold,

java.io.File repository)

构造函数

public void setSizeThreshold(int sizeThreshold)

设置内存缓冲区的大小，默认值为10K。当上传文件大于缓冲区大小时， fileupload组件将使用临时文件缓存上传文件。单位字节

public void setRepository(java.io.File repository)

指定临时文件目录

ServletFileUpload：负责处理上传的文件数据，并将表单中每个输入项封装成一个 FileItem 对象中。

常用方法有：

boolean isMultipartContent(HttpServletRequest request)

判断上传表单是否为multipart/form-data类型

List parseRequest(HttpServletRequest request)

解析request对象，并把表单中的每一个输入项包装成一个fileItem 对象，并返回一个保存了所有FileItem的list集合。

setFileSizeMax(long fileSizeMax)

设置上传文件的最大值

setSizeMax(long sizeMax)

设置上传文件总量的最大值

setHeaderEncoding(java.lang.String encoding)

设置编码格式

setProgressListener(ProgressListener pListener)

设置监听

执行步骤：

1. 创建DiskFileItemFactory对象

FileItemFactory factory = new DiskFileItemFactory();

1. 使用DiskFileItemFactory 对象创建ServletFileUpload对象

ServletFileUpload upload = new ServletFileUpload(factory);

1. 判断表单是否是multipart/form-data类型

ServletFileUpload.isMultipartContent(request)

4）调用parseRequest方法解析request对象，获取保存所有上传内容（FileItem对象）的List对象

5）对list进行迭代，调用其isFormField方法判断是否是上传文件。

6）设置存放文件的路径，保存文件

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

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

//创建文件上传的工厂类对象

FileItemFactory factory = **new** DiskFileItemFactory();

//根据工厂类对象创建ServletFileUpload对象

ServletFileUpload upload = **new** ServletFileUpload(factory);

//解决文件名的中文乱码问题

upload.setHeaderEncoding("utf-8");

//判断表单的类型是否是multpart/form-data

**if**(ServletFileUpload.*isMultipartContent*(request)){

**try** {

//将请求参数转换为FileItem对象的集合

List<FileItem> items = (List<FileItem>)upload.parseRequest(request);

**for** (FileItem fileItem : items) {

//判断是否是普通的表单控件数据

**if**(fileItem.isFormField()){

//得到表单中name属性的值

System.***out***.println(fileItem.getFieldName());

//得到表单参数的值

//指定编码方式，解决提交数据的中文乱码

System.***out***.println(fileItem.getString("utf-8"));

}**else**{

System.***out***.println(fileItem.getFieldName());

//得到文件名

System.***out***.println(fileItem.getName());

String path = "D:/upload";

File dir = **new** File(path);

//如果文件夹不存在，创建

**if**(!dir.exists()){

dir.mkdir();

}

String name = fileItem.getName();

Date date = **new** Date();

**long** time = date.getTime();

//得到文件的名称和后缀

String fileName = name.substring(0, name.lastIndexOf("."));

String ext = name.substring(name.lastIndexOf(".") + 1);

name = fileName + "\_" + time + "." + ext;

//创建上传文件的File对象

File file = **new** File(path, name);

//保存文件

fileItem.write(file);

//删除临时文件

fileItem.delete();

}

}

} **catch** (FileUploadException e) {

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

e.printStackTrace();

} **catch** (Exception e) {

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

e.printStackTrace();

}

}

# 二 文件下载

## 1 通过超链接

<a href="file/哈哈.rar">下载</a>

## 2 通过代码方式

本质上是通过流进行文件的拷贝，由于要下载的文件可以是任何类型，所以需要通过字节流进行处理

步骤：

1）获取要下载的文件的输入流

2）设置响应头的Content-Disposition的值

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

表示将请求所得的内容存为一个文件的时候提供一个默认的文件名

3）通过response得到输出流

4）通过IO流进行文件的拷贝

5）关闭流

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

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

String fileName = request.getParameter("name");

File file = **new** File("D:/upload", fileName);

FileInputStream in = **new** FileInputStream(file);

//如果文件名中包含中文，需要进行url编码

fileName = URLEncoder.*encode*(fileName, "utf-8");

//设置content-Disposition

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

//获取响应对象的输出流

ServletOutputStream outputStream = response.getOutputStream();

//文件拷贝

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

**int** len = -1;

**while**((len = in.read(buff)) != -1){

outputStream.write(buff, 0, len);

}

//关闭流

in.close();

outputStream.close();

}