# 1文件上传

## 1.1文件上传的原理分析

### 文件上传的前提

* 提供form表单，method必须是post
* form表单的enctype必须是**multipart/form-data**
* 提供**input type="file"**类的上传输入域

### enctype属性

**enctype作用**：告知服务器请求正文的MIME类型。（请求消息头：Content-Type作用是一致的）

**可选值：**

**application/x-www-form-urlencoded(默认)：**

正文：username=admin&password=123

服务器获取数据：String username = request.getParameter("username");

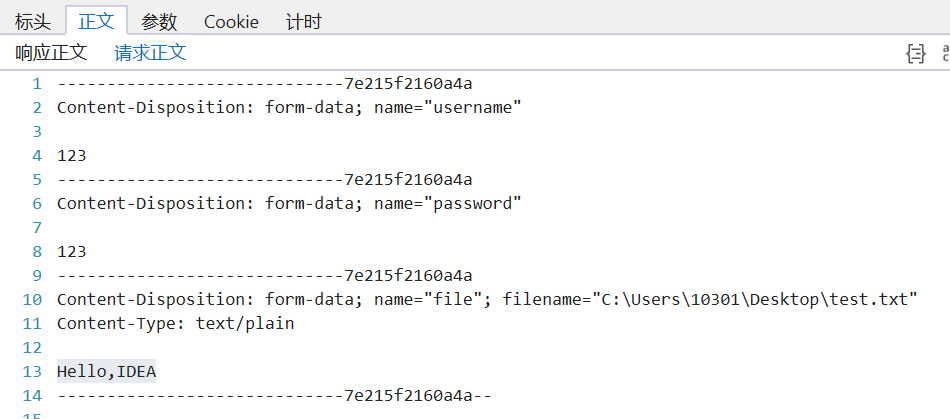
**multipart/form-data:**

正文：使用**IE的开发人员工具**来测试

服务器获取数据：request.getParameter(String)方法获取指定的表单字段字符内容，**但文件上传表单已经不在是字符内容，而是字节内容，所以失效**。

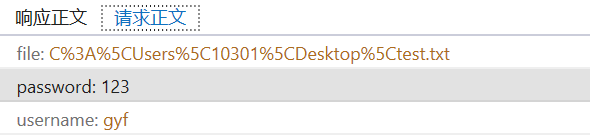
|  |
| --- |
|  |
|  |

multipart/form-data



application/x-www-form-urlencoded

username=gyf&password=123&file=c://///test.txt



request.getInputStream 获取表单的原始参数

## 1.2文件上传案例

### upload.jsp

|  |
| --- |
|  |

### UploadServlet

|  |
| --- |
|  |

## 1.3借用第三方解析文件

### fileupload概述

* fileupload是由apache的commons组件提供的上传组件。它最主要的工作就是帮我们解析request.getInputStream()。
* 使用fileupload只需要2个jar包

commons-fileupload.jar，核心包；

commons-io.jar，依赖包。

### fileupload的核心类有：

DiskFileItemFactory、ServletFileUpload、FileItem

### 解析原理

|  |
| --- |
|  |

### FileItem对象介绍

* FileItem对象对应一个表单项（表单字段）。可以是文件字段或普通字段
* boolean isFormField()：判断当前表单字段是否为普通文本字段，如果返回false，说明是文件字段；
* String getFieldName()：获取字段名称，例如：<input type=”text” name=”username”/>，返回的是username；
* String getString()：获取字段的内容，如果是文件字段，那么获取的是文件内容，当然上传的文件必须是文本文件；
* String getName()：获取文件字段的文件名称；（a.txt）
* String getContentType()：获取上传的**文件的MIME类型**，例如：text/plain。
* int **getSize()：获取上传文件的大小；**
* InputStream getInputStream()：获取上传文件对应的输入流；
* void write(File)：把上传的文件保存到指定文件中。
* delete():删除临时文件

### 使用fileupload实现文件上传代码

|  |
| --- |
| @WebServlet(**"/UploadServlet"**) **public class** UploadServlet **extends** HttpServlet{  @Override  **protected void** doPost(HttpServletRequest req, HttpServletResponse resp)  **throws** ServletException,IOException {   *//1.判断是否为文件上传表单* **if**(ServletFileUpload.*isMultipartContent*(req)){  *//2.创建个文件工厂* FileItemFactory factory = **new** DiskFileItemFactory();  *//3.创建文件上传对象* ServletFileUpload fileUpload = **new** ServletFileUpload(factory);   *//4.解析请求* **try**{  List<FileItem> items = fileUpload.parseRequest(req);  **for**(FileItem item : items){  **if**(item.isFormField()){*//普通表单字段* processFormField(item);  }**else**{*//文件字段* processFile(item);  }  }  }**catch** (FileUploadException fue){  fue.printStackTrace();  }   }   }   **private void** processFile(FileItem item) {  String fieldName = item.getFieldName();  String fieldValue = item.getString();  *//1.获取文件扩展名* String suffix = item.getName().split(**"\\."**)[1];  String saveFileName = UUID.*randomUUID*().toString() + **"."** + suffix;  System.***out***.println(saveFileName + **":"** + item.getContentType());   *//2.创建文件保存的文件夹* String folderPath = **this**.getServletContext().getRealPath(**"/WEB-INF/upload"**);  System.***out***.println(**"保存的路径:"** + folderPath);  File file = **new** File(folderPath);  **if**(!file.exists()){  file.mkdirs();  }   *//3.保存文件  //文件完整保存路径* String filePath = folderPath + **"/"** + saveFileName;  **try** {  item.write(**new** File(filePath));  item.delete();*//删除临时文件*  } **catch** (Exception e) {  e.printStackTrace();  }  }   **private void** processFormField(FileItem item) {  String fieldName = item.getFieldName();  String fieldValue = item.getString();  System.***out***.println(fieldName + **":"** + fieldValue);  } } |

## 1.4文件上传保存路径的几种方案

**方案一：添加日期目录，方便分类查找**

|  |
| --- |
|  |

**方式二：把日期20180108转成16进制字符串d12fde2f**

|  |
| --- |
|  |

## 1.5文件上传应该注意的问题

### 中文乱码问题

表单普通字段乱码用item.getString("UTF-8");

文件名乱码可用

|  |
| --- |
|  |

### 文件上传大小限制问题

单个文件大小：

ServletFileUpload.setFileSizeMax(字节)

|  |
| --- |
|  |

总文件大小：（多文件上传）

ServletFileUpload.setSizeMax(字节)

### 临时文件的问题

* DiskFileItemFactory:

作用：产生FileItem对象

内部有一个缓存，缓存大小默认是10Kb。如果上传的文件超过10Kb，用磁盘作为缓存。

存放缓存文件的目录在哪里？默认是系统的临时目录。

如果自己用IO流实现的文件上传，要在流关闭后，清理临时文件。调用FileItem.delete();

## 1.6 多文件上传

|  |
| --- |
|  |

**动态添加多文件上传**

### HTML

|  |
| --- |
| <**form action="${**pageContext.request.contextPath**}/UploadServlet" method="post" enctype="multipart/form-data"**>  用户名:<**input type="text" name="username" placeholder="请输入用户名"**><**br**>  密码：<**input type="password" name="password" placeholder="请输入密码"**><**br**>  <**div id="div1"**>  <**div**>  <**input type="file" name="photo"** /><**input type="button" value="添加" onclick="***addFile*()**"**/><**br** />  </**div**>  </**div**>  <**input type="submit" value="提交"**> </**form**> |

### JS

|  |
| --- |
| <**script type="text/javascript"**>  **function** *addFile*(){  *//得到div容器* **var** div1 = **document**.getElementById(**"div1"**);  div1.**innerHTML** += **"<div><input type='file' name='photo' /><input type='button' value='删除' onclick='***delFile***(this)'/><br /></div>"**;  }   **function** *delFile*(input){  *//使用input对象的爷爷删除他的爸爸* input.**parentNode**.**parentNode**.removeChild(input.**parentNode**);  } </**script**> |

# 2 文件下载

## 2.1文件下载需要设置响应头

### Content-Disposition

|  |  |
| --- | --- |
| attachment;filename=xx.png | 以附件形式下载 |

### Content-Type

|  |  |
| --- | --- |
| application/octet-stream | 二进制流 |

## 2.2 案例下载代码

|  |
| --- |
| @WebServlet(**"/DownloadServlet"**) **public class** DownloadServlet **extends** HttpServlet{  @Override  **protected void** doGet(HttpServletRequest req, HttpServletResponse resp)  **throws** ServletException, IOException {   *//1.文件下载的位置* String filePath = **"C:\\Users\\10301\\Desktop\\test\\598573506018647635.jpg"**;   *//广州国际金融中心  //2.文件名* String fileName = **"广州国际金融中心.jpg"**;  fileName = URLEncoder.*encode*(fileName,**"UTF-8"**);*//中文名文件需要进行URL编码  //3.设置响应头* resp.setHeader(**"Content-Disposition"**,**"attachment;filename="** + fileName);  resp.setContentType(**"application/octet-stream"**);   *//4.响应文件* File file = **new** File(filePath);  InputStream is = **new** FileInputStream(file);  **byte**[] bytes = **new byte**[1024 \* 8];  **int** len;  OutputStream os = resp.getOutputStream();  **while**((len = is.read(bytes)) != -1){  os.write(bytes,0,len);  }   *//5.关闭流* is.close();  os.close();  } } |