# 1. 文件上传与下载

## 1.1 文件上传

### 文件上传，要点:

前台：

1. 提交方式：post

2. 表单中有文件上传的表单项： <input type=”file” />

3. 指定表单类型:

默认类型：enctype="application/x-www-form-urlencoded"

**文件上传类型：multipart/form-data**

### 手动实现文件上传

|  |
| --- |
| <form method=*"post"* enctype=*"multipart/form-data"* action=*"*${pageContext.request.contextPath}*/day12/Upload"*>  用户名：<input type=*"text"* name=*"username"* /><br/>  上传文件：<input type=*"file"* name=*"file1"*/><br/>  <input type=*"submit"* value=*"上传"* />  </form> |

|  |
| --- |
|  |
| **public** **void** doPost(HttpServletRequest request, HttpServletResponse response)  **throws** ServletException, IOException {    ServletInputStream in = request.getInputStream();    BufferedReader reader=**new** BufferedReader(**new** InputStreamReader(in));    String str=**null**;  **while**((str=reader.readLine())!=**null**)  {  System.***out***.println(str);  }    reader.close();  in.close();  } |
| 输出结果： |
| -----------------------------7e0e536110d82  Content-Disposition: form-data; name="username"  John  -----------------------------7e0e536110d82  Content-Disposition: form-data; name="file1"; filename="C:\Users\liapbook\Desktop\系统搭建.txt"  Content-Type: text/plain  ﻿外网访问公司OA，http://122.224.233.66:3189/oa  http://122.224.233.66:3188/oa  公司Exmobi：http://122.224.233.66:394/  192.168.3.189/oa/Extensions\InvokeCenter\WebServices\OA\OAWebServices.asmx  192.168.3.189/oa/Extensions\InvokeCenter\WebServices\Framework\FrameworkWebServices.asmx  -----------------------------7e0e536110d82-- |
| 总结：  最终获取数据，要对上面的结果进行解析！  文件上传，在开发中经常用，每次都写解析程序！(工具类)  也可以使用开源的文件上传组件-FileUpload组件！ |
|  |

### Apache提供的文件上传组件:FileUpload组件

文件上传功能开发中比较常用，apache也提供了文件上传组件！

FileUpload组件:

1. 下载源码

2. 项目中引入jar文件

commons-fileupload-1.2.1.jar 【文件上传组件核心jar包】

commons-io-1.4.jar 【封装了对文件处理的相关工具类】

组件API:

|-------FIleItemFactory 文件上传工厂类（把请求中的表单元素封装成一个个FileItem）

setRepository 设置文件上传临时目录

|-------ServletFileUpload 文件上传核心类，可以获取所有的Item对象

parseRequest 获取所有item,和request.getParamiter方法冲突，重新组装了request

isMultipartContent 判断表单是否是上传表单

setFileSizeMax 设置单个文件的上传最大值

setSizeMax 设置单个表单最大上传文件大小

|--------FileItem

getString 获取数据

getString(“UTF-8”) 获取

write 写文件到指定目录

delete 删除临时文件

|  |
| --- |
| /\*\*  \* 1.创建文件上传工厂类  \*/  FileItemFactory fc= **new** DiskFileItemFactory();  /\*\*  \* 2.创建文件上传核心类对象  \*/  ServletFileUpload upload = **new** ServletFileUpload(fc);  //判断表单是否为上传表单  **if**(upload.*isMultipartContent*(request))  {  **try** {  /\*\*  \* 3.把请求数据转换成FileItem对象的集合  \*/  List<FileItem> fileitems = upload.parseRequest(request);  **for**(FileItem item : fileitems){  //假如是普通表单项  **if**(item.isFormField())  {  String fieldName = item.getFieldName();//普通元素的name值  String value = item.getString();//普通元素的值  System.***out***.println(fieldName + ":" + value);  }  **else**{  String contentType = item.getContentType();//文件类型  String fileName = item.getName().substring(item.getName().lastIndexOf('\\'), item.getName().length());//文件名  InputStream in = item.getInputStream();//文件流    BufferedInputStream bin=**new** BufferedInputStream(in);  FileOutputStream fos = **new** FileOutputStream("D:\\"+fileName);  BufferedOutputStream bos = **new** BufferedOutputStream(fos);    **int** len = 0;  **while** ((len = bin.read()) != -1)  {  bos.write(len);  }  bin.close();  bos.close();  }  }    } **catch** (FileUploadException e) {  e.printStackTrace();  }  }  **else**{  System.***out***.println("当前表单不是文件上传表单，处理失败！");  } |

### 文件上传与下载，完整案例：

步骤：

1. 文件上传

2. 列表下载

|  |
| --- |
| Index.jsp |
| <body>  <a href=*"*${pageContext.request.contextPath }*/upload.jsp"*>文件上传</a> &nbsp;&nbsp;&nbsp;  <a href=*"*${pageContext.request.contextPath }*/fileServlet?method=downList"*>文件下载</a>    </body> |
| Upload.jsp |
| <body>  <form name=*"frm\_test"* action=*"*${pageContext.request.contextPath }*/fileServlet?method=upload"* method=*"post"* enctype=*"multipart/form-data"*>  <%--<input type="hidden" name="method" value="upload">--%>    用户名：<input type=*"text"* name=*"userName"*> <br/>  文件： <input type=*"file"* name=*"file\_img"*> <br/>    <input type=*"submit"* value=*"提交"*>  </form>  </body> |
| Downlist.jsp |
| <body>  <table border=*"1"* align=*"center"*>  <tr>  <th>序号</th>  <th>文件名</th>  <th>操作</th>  </tr>  <c:forEach var=*"en"* items="${requestScope.fileNames}" varStatus=*"vs"*>  <tr>  <td>${vs.count }</td>  <td>${en.value }</td>  <td>  <%--<a href="${pageContext.request.contextPath }/fileServlet?method=down&..">下载</a>--%>  <!-- 构建一个地址 -->  <c:url var=*"url"* value=*"fileServlet"*>  <c:param name=*"method"* value=*"down"*></c:param>  <c:param name=*"fileName"* value="${en.key}"></c:param>  </c:url>  <!-- 使用上面地址 -->  <a href=*"*${url }*"*>下载</a>  </td>  </tr>  </c:forEach>  </table>  </body> |
| FileServlet.java |
| /\*\*  \* 处理文件上传与下载  \* **@author** zhanghong  \*  \*/  **public** **class** FileServlet **extends** HttpServlet {  **public** **void** doGet(HttpServletRequest request, HttpServletResponse response)  **throws** ServletException, IOException {  // 获取请求参数： 区分不同的操作类型  String method = request.getParameter("method");  **if** ("upload".equals(method)) {  // 上传  upload(request,response);  }    **else** **if** ("downList".equals(method)) {  // 进入下载列表  downList(request,response);  }    **else** **if** ("down".equals(method)) {  // 下载  down(request,response);  }  }      /\*\*  \* 1. 上传  \*/  **private** **void** upload(HttpServletRequest request, HttpServletResponse response)  **throws** ServletException, IOException {    **try** {  // 1. 创建工厂对象  FileItemFactory factory = **new** DiskFileItemFactory();  // 2. 文件上传核心工具类  ServletFileUpload upload = **new** ServletFileUpload(factory);  // 设置大小限制参数  upload.setFileSizeMax(10\*1024\*1024); // 单个文件大小限制  upload.setSizeMax(50\*1024\*1024); // 总文件大小限制  upload.setHeaderEncoding("UTF-8"); // 对中文文件编码处理  // 判断  **if** (upload.*isMultipartContent*(request)) {  // 3. 把请求数据转换为list集合  List<FileItem> list = upload.parseRequest(request);  // 遍历  **for** (FileItem item : list){  // 判断：普通文本数据  **if** (item.isFormField()){  // 获取名称  String name = item.getFieldName();  // 获取值  String value = item.getString();  System.*out*.println(value);  }  // 文件表单项  **else** {  /\*\*\*\*\*\*\*\* 文件上传 \*\*\*\*\*\*\*\*\*\*\*/  // a. 获取文件名称  String name = item.getName();  // ----处理上传文件名重名问题----  // a1. 先得到唯一标记  String id = UUID.*randomUUID*().toString();  // a2. 拼接文件名  name = id + "#" + name;    // b. 得到上传目录  String basePath = getServletContext().getRealPath("/upload");  // c. 创建要上传的文件对象  File file = **new** File(basePath,name);  // d. 上传  item.write(file);  item.delete(); // 删除组件运行时产生的临时文件  }  }  }  } **catch** (Exception e) {  e.printStackTrace();  }      }    /\*\*  \* 2. 进入下载列表  \*/  **private** **void** downList(HttpServletRequest request, HttpServletResponse response)  **throws** ServletException, IOException {    // 实现思路：先获取upload目录下所有文件的文件名，再保存；跳转到down.jsp列表展示    //1. 初始化map集合Map<包含唯一标记的文件名, 简短文件名> ;  Map<String,String> fileNames = **new** HashMap<String,String>();    //2. 获取上传目录，及其下所有的文件的文件名  String bathPath = getServletContext().getRealPath("/upload");  // 目录  File file = **new** File(bathPath);  // 目录下，所有文件名  String list[] = file.list();  // 遍历，封装  **if** (list != **null** && list.length > 0){  **for** (**int** i=0; i<list.length; i++){  // 全名  String fileName = list[i];  // 短名  String shortName = fileName.substring(fileName.lastIndexOf("#")+1);  // 封装  fileNames.put(fileName, shortName);  }  }    // 3. 保存到request域  request.setAttribute("fileNames", fileNames);  // 4. 转发  request.getRequestDispatcher("/downlist.jsp").forward(request, response);  }    /\*\*  \* 3. 处理下载  \*/  **private** **void** down(HttpServletRequest request, HttpServletResponse response)  **throws** ServletException, IOException {    // 获取用户下载的文件名称(url地址后追加数据,get)  String fileName = request.getParameter("fileName");  fileName = **new** String(fileName.getBytes("ISO8859-1"),"UTF-8");    // 先获取上传目录路径  String basePath = getServletContext().getRealPath("/upload");  // 获取一个文件流  InputStream in = **new** FileInputStream(**new** File(basePath,fileName));    // 如果文件名是中文，需要进行url编码  fileName = URLEncoder.*encode*(fileName, "UTF-8");  // 设置下载的响应头  response.setHeader("content-disposition", "attachment;fileName=" + fileName);    // 获取response字节流  OutputStream out = response.getOutputStream();  **byte**[] b = **new** **byte**[1024];  **int** len = -1;  **while** ((len = in.read(b)) != -1){  out.write(b, 0, len);  }  // 关闭  out.close();  in.close();      }    **public** **void** doPost(HttpServletRequest request, HttpServletResponse response)  **throws** ServletException, IOException {  **this**.doGet(request, response);  }  } |