# 文件的上传和下载servlet端

**依赖**javax.servlet-api 3.1**的文件上传**

maven**依赖此**jar

```
@WebServlet( "/javaUpload" )
@MultipartConfig
public class javaUpload extends BaseServlet {
 private static final long serialVersionUID = 1L;
 protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
 forward("webuploader", req, resp);
 }
 protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

 Part part = req.getPart("file");
 //获得文件 名 大小 类型
 String header = part.getHeader("Content-Disposition");

 // 拼接上传的文件名 生成唯一的文件名
 header = header.split(";")[2].split("\"")[1];

 String fileName = UUID.randomUUID().toString()+header.substring(header.lastIndexOf("."));
 //从part获得文件的输入流
 InputStream in = part.getInputStream();
 //新建输出流，在文件夹下的文件名
 OutputStream out = new FileOutputStream(new File(new File("D:/upload3"), fileName));
 //从part流中数据copy到指定文件中
 IOUtils.copy(in, out);
 Map<String ,Object> map = new HashMap();
 //目前只发送 上传成功后的json 响应数据
 map.put("success", true);//富文本框文件上传图片成功的json信号（浏览器端富文本框发现此信号会自动把file_path 对应信息展示
 map.put("file_path", Config.getProperty("http.address")+ fileName);//上传成功后，向前端传送此图片的请求资源路径名
 map.put("fileName", fileName);//向前端传送根据此图片生成的图片名
 send(resp, map);//以json数据响应前端
 }
}
```

Apache  commons-fileupload

**依赖**commons-fileupload **包**1.3.1

```
@WebServlet("/upload")
public class Upload extends BaseServlet {
	private static final long serialVersionUID = 1L;
	@Override
	protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
		forward("text/upload", req, resp);
	}
	@Override
	protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
		req.setCharacterEncoding("UTF-8");
		// 1.设置文件上传路径
		File upFile = new File("D:/upload");
		if (!upFile.exists()) {
			upFile.mkdir();
		}
		// 2、设置临时路径
		File tempFile = new File("D:/temp");
```

```
        if (!tempFile.exists()) {
            tempFile.mkdir();
        }
        // 3.判断表单属性 enctype 是否为multipart/form-data
        if (ServletFileUpload.isMultipartContent(req)) {

            DiskFileItemFactory factory = new DiskFileItemFactory();

            factory.setSizeThreshold(1024);

            factory.setRepository(tempFile);
            // 根据factory 创建servlet文件上传对象
            ServletFileUpload upload = new ServletFileUpload(factory);
            // 用对象获得form表单 数据的集合

            try {
                List<FileItem> itemList = upload.parseRequest(req);
                for (FileItem item : itemList) {
                    if (item.isFormField()) {
                        // 不同数据获得此数据的信息
                        String fileName = item.getFieldName();
                        String value = item.getString("UTF-8");
                    } else {
                        // 为文件数据时
                        String fileName = item.getFieldName();
                        String name = item.getName();
                        // 生成唯一的文件名
                        name = UUID.randomUUID().toString() + name.substring(name.lastIndexOf("."));
                        InputStream in = item.getInputStream();
                        FileOutputStream out = new FileOutputStream(new File(upFile, name));
                        IOUtils.copy(in, out);
                        out.flush();
                        out.close();
                        in.close();
                        System.out.println("文件上传成功！");

                    }
                }

            } catch (FileUploadException e) {
                e.printStackTrace();
            }
        } else {
            throw new ServletException("表单异常");
        }
    }
}
```

**文件的预览和下载**

```
@WebServlet("/download")
public class Download extends BaseServlet {
    private static final long serialVersionUID = 1L;

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        // 下载的文件名，在get 请求中
        String fileName = req.getParameter("fileName");//获得请求中参数UUID 文件名
        String name = req.getParameter("name");//获得文件中文名
        //下载的资源路径文件
        File file = new File(new File("D:/img/upload"), fileName);
        if (file.exists()) {
            //下载的配置 前端根据响应头决定下载和预览
            if (StringUtils.isNotEmpty(name)) {
                // 设置文件类型
                resp.setContentType("application/octet-stream");
                // 告知文件大小
                resp.setContentLength((int) file.length());
                // 添加响应头 传入前端
                resp.addHeader("Content-Disposition", "attachment; filename=\"" + name + "\"");
            }
            //从file中获得输入流  输出到响应流
            InputStream in = new FileInputStream(file);
            OutputStream out = resp.getOutputStream();
      //
            IOUtils.copy(in, out);
            out.flush();
            out.close();
            in.close();
        } else {
            resp.sendError(404, "资源未找到");
        }
    }
}
```