

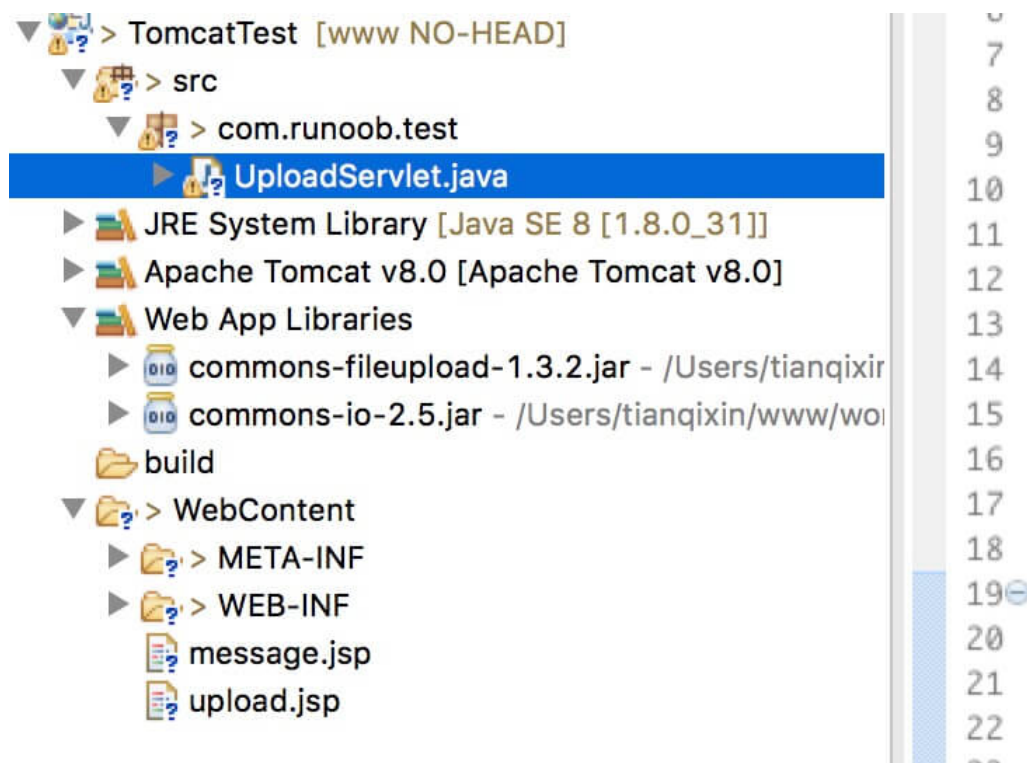
# Servlet 文件上传

Servlet 可以与 HTML 表单标签一起使用，来允许用户上传文件到服务器。上传的文件可以是文本文件或图像文件或任何文档。

本文使用到的文件有：

- upload.jsp: 文件上传表单。
- message.jsp: 提交成功后替换页面。
- UploadServlet.java: 上传处理 Servlet。
- 需要约会的 jar 文件: commons-fileupload-1.3.2, commons-io-2.5.jar。

结构图如下所示：



**注意：** Servlet3.0 已经内置了文件上传这一特性，开发者不再需要将 Commons FileUpload 组件引入到工程中去。

接下来我们详细介绍。

## 创建一个文件上传表单

下面的 HTML 代码创建了一个文件上传表单。以下几点需要注意：

- 表单方法属性应该设置为 **POST** 方法，不能使用 GET 方法。
- 表单 **enctype** 属性应该设置为 **multipart / form-data**。
- 表单动作属性应该设置为在服务器上方处理文件上传的 Servlet 文件。下面的实例使用了 **UpServlet** Servlet 来上传文件。
- 添加副本文件，您应该使用多个带有属性 **type = "file"** 的 `<input ... />` 标签。为了允许多个文件上传，请包含多个名称属性值不同的输入标签。输入标签具有不同的名称属性的值。浏览器会为每个输入标签关联一个浏览按钮。

upload.jsp 文件代码如下：

```
<%@页面语言=" java" contentType = " text / html; charset = UTF-8"
    pageEncoding = " UTF-8"%>
<!DOCTYPE html PUBLIC "-// W3C // DTD HTML 4.01 Transitional // EN"
    " http://www.w3.org/TR/html4/loose.dtd">
<html>
<头>
<meta http-equiv = " Content-Type" content = " text / html; charset = UTF-8">
<title>文件上传实例-菜鸟教程</ title>
</ head>
<身体>
<h1>文件上传实例-菜鸟教程</ h1>
<form method = " post" action = " / TomcatTest / UploadServlet" enctype = " multipart / form-data">
    选择一个文件:
    <input type = " file" name = " uploadFile" />
    <br/> <br/>
    <input type = " submit" value = "上传" />
</ form>
</ body>
</ html>
```

## 编写后台 Servlet

以下是 UploadServlet 的源代码，同于处理文件上传，在这之前我们先确保依赖包已经约会到项目的 WEB-INF

lib 目录下：

- 下面的实例依赖于 FileUpload，所以一定要确保在您的 classpath 中有最新版本的 **commons-fileupload.x** **xjar** 文件。可以从 <http://commons.apache.org/proper/commons-fileupload/> 下载。
- FileUpload 依赖于 Commons IO，所以一定要确保在您的 classpath 中有最新版本的 **commons-io-xxjar** 文件。可以从 <http://commons.apache.org/proper/commons-io/> 下载。

你可以直接下载本站提供的两个依赖包：

- [commons-fileupload-1.3.2.jar](#)
- [commons-io-2.5.jar](#)

UploadServlet 的源代码如下所示:

```
包 com.runoob.test;

导入 java.io.File;
导入 java.io.IOException;
导入 java.io.PrintWriter;
导入 java.util.List;

导入 javax.servlet.ServletException;
导入 javax.servlet.annotation.WebServlet;
导入 javax.servlet.http.HttpServlet;
导入 javax.servlet.http.HttpServletRequest;
导入 javax.servlet.http.HttpServletResponse;

导入 org.apache.commons.fileupload.FileItem;
导入 org.apache.commons.fileupload.disk.DiskFileItemFactory;
导入 org.apache.commons.fileupload.servlet.ServletFileUpload;

/ **
 * Servlet 实现类 UploadServlet
 * /
@WebServlet (“ / UploadServlet”)
公共类 UploadServlet 扩展了 HttpServlet {
    私有静态最终长 serialVersionUID = 1L;

    //上传文件存储目录
    私有静态最终字符串 UPLOAD_DIRECTORY =“ upload”;

    //上传配置
    私有静态最终 int MEMORY_THRESHOLD = 1024 * 1024 * 3; // 3MB
    私有静态最终整数 MAX_FILE_SIZE = 1024 * 1024 * 40; // 40MB
    私有静态最终整数 MAX_REQUEST_SIZE = 1024 * 1024 * 50; // 50MB

/ **
 *发布数据及保存文件
 * /
受保护的 void doPost (HttpServletRequest 请求,
    HttpServletResponse 响应) 引发 ServletException, IOException {
    //检测是否为多媒体上传
    如果 (! ServletFileUpload.isMultipartContent (request) ) {
```

```
//如果不是则停止
PrintWriter writer = response.getWriter ();
writer.println (“错误: 表单必须包含 enctype = multipart / form-data”);
writer.flush ();
返回;
}

//配置上传参数
DiskFileItemFactory factory = new DiskFileItemFactory ();
//设置内存临界值-超过后将产生临时文件并存储到临时目录中
factory.setSizeThreshold (MEMORY_THRESHOLD);
//设置临时存储目录
factory.setRepository (new File (System.getProperty (“ java.io.tmpdir”))) );

ServletFileUpload upload = new ServletFileUpload (factory);

//设置最大文件上传值
upload.setFileSizeMax (MAX_FILE_SIZE);

//设置最大请求值 (包含文件和表单数据)
upload.setSizeMax (MAX_REQUEST_SIZE);

//中文处理
upload.setHeaderEncoding (“ UTF-8”);

//构造临时路径来存储上传的文件
//这个路径相对当前应用的目录
字符串 uploadPath = request.getServletContext ().getRealPath (“ ./”) + File.separator + UP
LOAD_DIRECTORY;

//如果目录不存在则创建
文件 uploadDir =新文件 (uploadPath);
如果 (! uploadDir.exists ()) {
    uploadDir.mkdir ();
}

尝试{
    //解析请求的内容提取文件数据
    @SuppressWarnings (“未选中”)
    List <FileItem> formItems = upload.parseRequest (request);

    如果 (formItems != null && formItems.size () > 0) {
        //继承表单数据
```

```
    对于 (FileItem item: formItems) {
        //处理不在表单中的细分
        如果 (! item.isFormField () ) {
            字符串 fileName = new File (item.getName () ) 。getName () ;
            字符串 filePath = uploadPath + File.separator + fileName;
            文件 storeFile = new File (filePath) ;
            //在控制台输出文件的上传路径
            System.out.println (filePath) ;
            //保存文件到硬盘
            item.write (storeFile) ;
            request.setAttribute (" message",
                "文件上传成功! ") ;
        }
    }
}
} catch (异常例外) {
    request.setAttribute (" message",
        "错误信息: " + ex.getMessage () ) ;
}
//反弹到 message.jsp
request.getRequestDispatcher (" / message.jsp") 。forward (
    请求, 回应) ;
}
}
```

message.jsp 文件代码如下:

```
<%@页面语言=" java" contentType = " text / html; charset = UTF-8"
    pageEncoding = " UTF-8"%>
<! DOCTYPE html PUBLIC "-// W3C // DTD HTML 4.01 Transitional // EN"
    " http://www.w3.org/TR/html4/loose.dtd">
<html>
<头>
<meta http-equiv = " Content-Type" content = " text / html; charset = UTF-8">
<title>文件上传结果</ title>
</ head>
<身体>
    <中心>
        <h2> $ {message} </ h2>
    </ center>
</ body>
</ html>
```

## 编译和运行 Servlet

编译上面的 Servlet UploadServlet，并在 web.xml 文件中创建所需的位置，如下所示：

```
<? xml 版本=" 1.0"编码=" UTF-8"? >
<web-app xmlns: xsi =" http://www.w3.org/2001/XMLSchema-instance"
  xmlns =" http://java.sun.com/xml/ns/javaee"
  xmlns: web =" http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  xsi: schemaLocation =" http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  id =" WebApp_ID"版本=" 2.5">
<servlet>
  <display-name> UploadServlet </ display-name>
  <servlet-name> UploadServlet </ servlet-name>
  <servlet-class> com.runoob.test.UploadServlet </ servlet-class>
</ servlet>

<servlet 映射>
  <servlet-name> UploadServlet </ servlet-name>
  <url-pattern> / TomcatTest / UploadServlet </ url-pattern>
</ servlet-mapping>
</ web-app>
```

现在尝试使用您在上面创建的 HTML 表单来上传文件。当您在浏览器中访问：<http://localhost:8080/TomcatTest/upload.jsp>，演示如下所示：

