

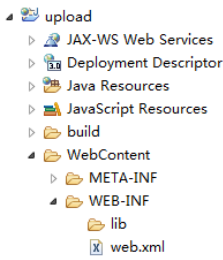
小兔Java教程 - 三分钟学会Java文件上传



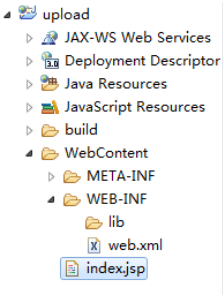
今天群里正好有人问起了Java文件上传的事情，本来这是Java里面的知识点，而我目前最主要的精力还是放在了JS的部分。不过反正也不麻烦，我就专门开一贴来聊聊Java文件上传的基本实现方法吧。

话不多说，直接开始。

第一步，新建一个空白的web项目。开发工具我还是用的eclipse，我不喜欢用IDEAL。



接下来，编写index.jsp



```
<%@ page language="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>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Java文件上传</title>
</head>
<body>

</body>
</html>
```

body里面就简单弄一个上传框了，上传的服务器地址为service/upload.jsp，也就是WebContent目录下，一个叫做service文件夹里面的upload.jsp。

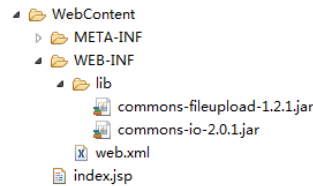
JSP就是Servlet，也就是所谓的服务器程序。它既可以做表现层，也可以做服务层。

代码：

```
<h1>文件上传</h1>

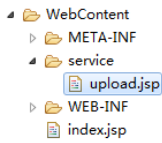
<h1>文件上传</h1>
<form action="service/upload.jsp" method="post" enctype="multipart/form-data">
    <input type="file" name="file">
    <input type="submit" value="文件上传">
</form>
```

接下来，导入两个jar包，分别是 **common-fileupload-1.2.jar** 和 **commons-io-2.0.1.jar**。

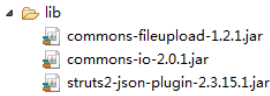


第二步，编写文件上传的服务类，也就是服务器端了。（我就直接写JSP了，懒得写Servlet）

先在WebContent目录下增加一个service文件夹,再在里面新建一个upload.jsp。



为了看一下效果，我们再导入一个操作JSON的jar包。



然后，在JSP页面导包：

```
<%@page import="org.apache.struts2.json.JSONUtil"%>
<%@page import="java.io.File"%>
<%@page import="org.apache.commons.fileupload.FileItem"%>
<%@page import="org.apache.commons.fileupload.servlet.ServletFileUpload"%>
<%@page import="org.apache.commons.fileupload.disk.DiskFileItemFactory"%>
<%@page import="org.apache.commons.fileupload.FileItemFactory"%>
<%@page import="java.text.DecimalFormat"%>
<%@ page language="java" import="java.util.*" pageEncoding="UTF-8"%>
```

接下来，要在JSP页面写Java代码，先弄一对尖括号，Java代码就写在里面：

```
<%
%>
```

01、获取文件上传的真实路径（就是你要把文件保存在项目的什么地方？）

```
//获取文件的上传的具体目录，就是你的web项目的真实路径
String realPath = request.getSession().getServletContext().getRealPath("/");
//定义上传的目录
String dirPath = realPath+"/upload";
File dirFile = new File(dirPath);
```

02、如果该路径不存在就自动创建

```
//自动创建上传的目录
if(!dirFile.exists())dirFile.mkdirs();
```

我们把上传的文件放在WebContent目录下，也就是项目根路径下的一个叫做upload的文件夹内，如果没有，就自动创建一个。

03、进行上传操作，文件名用UUID

```
//上传操作
FileItemFactory factory = new DiskFileItemFactory();
ServletFileUpload upload = new ServletFileUpload(factory);
String fileName = null;
HashMap<String,Object> map = new HashMap<String,Object>();
try{
    List items = upload.parseRequest(request);
    if(null != items){
        Iterator itr = items.iterator();
        while(itr.hasNext()){
            FileItem item = (FileItem)itr.next();
            if(item.isFormField()){
                continue;
            }else{
                fileName = UUID.randomUUID().toString()+getExt(item.getName());
                //上传文件的目录
                File savedFile = new File(dirPath,fileName);
                item.write(savedFile);
                map.put("name",item.getName()); //文件的重视名称
                map.put("size",item.getSize()); //文件的真实大小
                map.put("sizeString",countFileSize(item.getSize())); //获取文件转换以后的大写
                map.put("url","upload/"+fileName); //获取文件的具体服务器的目录
            }
        }
    }
}catch(Exception e){
    e.printStackTrace();
}
```

最后，在新的页面打印文件的具体信息：

```
out.print(JSONUtil.serialize(map));
```

辅助函数：

```
<%!
/**
 * 文件大小转换成格式化的字符串
 */
public static String countFileSize(long fileSize) {
    String fileSizeString = "";
    try {
        DecimalFormat df = new DecimalFormat("#.00");
        long fileS = fileSize;
        if (fileS == 0) {
            fileSizeString = "0KB";
        } else if (fileS < 1024) {
            fileSizeString = df.format((double) fileS) + "B";
        } else if (fileS < 1048576) {
            fileSizeString = df.format((double) fileS / 1024) + "KB";
        } else if (fileS < 1073741824) {
            fileSizeString = df
                .format(((double) fileS / 1024 / 1024) - 0.01)
                + "MB";
        } else {
            fileSizeString = df.format((double) fileS / 1024 / 1024 / 1024)
                + "G";
        }
    } catch (Exception e) {
```

```

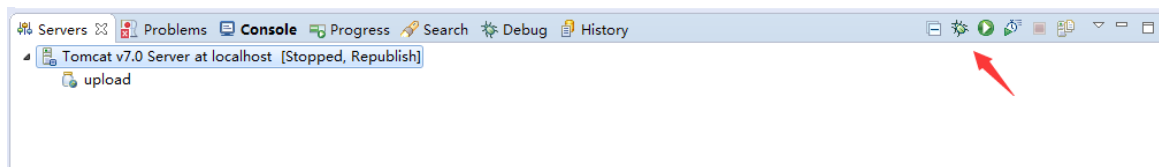
        e.printStackTrace();
    }
    return fileSizeString;
}

/**
 * 获取一个文件的后缀 (带有点)
 *
 * @param fileName
 *      文件名
 * @return 返回文件的后缀
 */
public static String getExt(String fileName) {
    int pos = fileName.lastIndexOf(".");
    if (pos == -1)
        return "";
    return fileName.substring(pos, fileName.length());
}

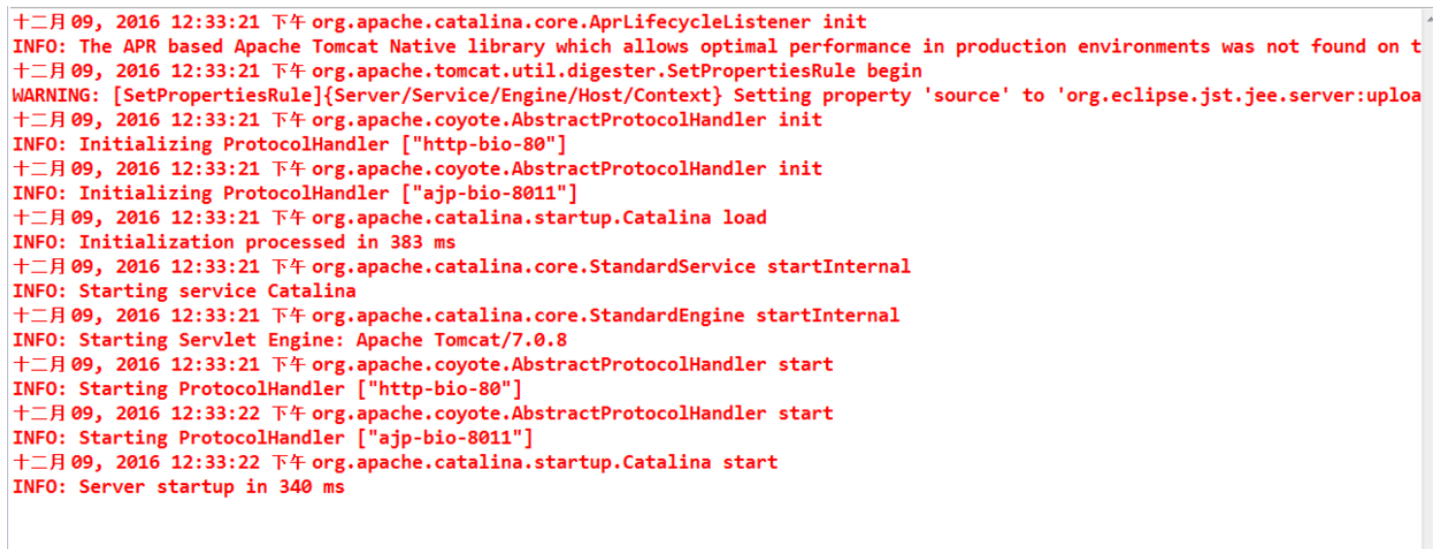
/**
 * 获取一个文件的后缀 (不带有点)
 *
 * @param fileName
 *      文件名
 * @return 返回文件的后缀
 */
public static String getExtNoPoint(String fileName) {
    if (fileName.lastIndexOf(".") == -1)
        return "";
    int pos = fileName.lastIndexOf(".") + 1;
    return fileName.substring(pos, fileName.length());
}
}
%>

```

OK，现在启动tomcat服务器。



没报错。



因为在web.xml里面配置了欢迎页，所以直接访问。

<http://localhost/upload>

文件上传

选择文件 未选择任何文件 文件上传

随便上传一张图片，点击文件上传，

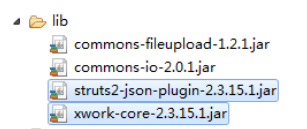
```

java.lang.ClassNotFoundException: com.opensymphony.xwork2.util.logging.LoggerFactory
    at org.apache.catalina.loader.WebappClassLoader.loadClass(WebappClassLoader.java:1676)
    at org.apache.catalina.loader.WebappClassLoader.loadClass(WebappClassLoader.java:1521)
    at org.apache.struts2.json.JSONUtil.<clinit>(JSONUtil.java:57)
    at org.apache.jsp.service.upload_jsp._jspService(upload_jsp.java:165)
    at org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:70)
    ...

```

报错了，看样子还是缺少jar包。

缺了xwork-core-2.3.15.1.jar 这个包，貌似这两包



是有依赖关系的，其实，这里只是为了把文件信息转成JSON格式，如果你有其他好用的jar包的话，也可以，不一定非要用这两个jar包。

导进去，重启tomcat，上传文件，就OK了。

打印出来的信息：

```
{"name":"'1929342-c2daab1a3c58006d.jpg","sizeString":'564.08KB',"url":"'upload/b4a08821-3d1a-41e6-b6e4-7985487513b0.jpg","size":577614}
```

再去发布目录看看

果然，它创建了一个upload文件夹：

名称	修改日期	类型	大小
META-INF	2016/12/9 12:33	文件夹	
service	2016/12/9 12:43	文件夹	
upload	2016/12/9 12:44	文件夹	
WEB-INF	2016/12/9 12:33	文件夹	
index.jsp	2016/12/9 12:13	JSP 文件	1 KB

图片也成功上传了~



好的，本文就介绍到这里，至于图片的真实路径，一般来说你要保存到数据库里面的。我个人不建议通过IO流的方式直接把图片保存到数据库，如果项目很大，你的数据库动辄几百个G，有点吃不消的。还是把图片放在硬盘上吧，如果要分布式，再另想办法，比如那个ngx啥的。

源码地址：<http://pan.baidu.com/s/1b7E1oa>
个人博客：<http://www.xiaotublog.com/index.html>