

Java Web 开发中文件上传方法研究与实现

陈涛, 黄艳峰

(商丘师范学院计算机与信息技术学院, 河南 商丘 476000)

摘要: 文件上传是 Web 开发中经常需要实现的功能, 介绍了 Java Web 开发中主流的文件上传方法, 包括 SmartUpload 和 Commons FileUpload 两种组件上传, 以及 Servlet 3.0 支持的原生 API 文件上传。给出了三种方法实现文件上传的主要代码, 并对三者上传大文件时的效率进行了比较。

关键词: SmartUpload; Commons FileUpload; Servlet 3.0; 文件上传; Java Web

中图分类号: TP311 文献标识码: A 文章编号: 1009-3044(2016)11-0048-02

DOI: 10.14004/j.cnki.ckt.2016.1265

在 Web 应用开发中, 文件上传是经常需要实现的功能, 目前广为流传的方法是使用第三方的文件上传组件。具体在 Java web 开发领域, 常用的上传组件有 SmartUpload、Apache Commons FileUpload 等。在最新的 Servlet 3.0 规范中, 对文件上传提供了 API 的原生支持, 这样就可以不再依赖任何第三方上传组件。本文将举例探讨以上三种文件上传方法的特点, 最后从上传效率上给出三种方法的比较。

1 SmartUpload 组件

SmartUpload 是在 Java Web 开发中使用较多的一个文件上传下载组件, 它的特点是使用非常简单, 只需几行代码即可实现文件的上传下载, 所以即使 SmartUpload 已经停止了更新, 但直到今天依然是很多开发者的选择。

SmartUpload 的 API 中主要包括 SmartUpload、Files、File、Request 等核心类。由于篇幅限制, 本文就不再列出各个类的方法了, 读者可自行上网查阅。下面举例说明该组件的使用方法。

1) 编写文件上传表单文件

文件上传表单使用属性值为 file 的 input 标记, 注意表单必须以 post 方式提交, 表单的 enctype 属性必须是 multipart/form-data。该文件在三种文件上传方法中都需要用到。

```
<html>
<head><title>SmartUpload 文件上传</title></head>
<body>
<form action="upload.jsp" method="post" enctype="multipart/form-data">
```

选择要上传的文件: <input type="file" name="filename">

<input type="submit" value="上传">

</form></body></html>

2) 编写 JSP 文件实现文件上传

首先需要将下载的 SmartUpload 组件 jar 包拷贝至 Web 应用的 WEB-INF\lib 目录下, 然后编写实现文件上传功能的 JSP

文件。文件上传核心代码如下:

```
SmartUpload su= new SmartUpload();//创建 SmartUpload 对象
su.initialize(pageContext); //初始化 SmartUpload 对象
su.setMaxFileSize(300*1024*1024); //设置单个文件最大上传大小, 如 300M
su.setAllowedFilesList("avi,wmv"); //设置允许的上传文件类型
try {
    su.upload(); //执行上传
    File file = su.getFiles().getFile(0); //获取第一个上传文件, 若上传多个则循环获取
    String savePath = "upload\\" + file.getFileName(); //定义上传文件的保存路径
    file.saveAs(attach_url_path, File.SAVEAS_VIRTUAL); //保存上传文件到指定路径
} catch (Exception e) { //可能出现违反上传文件大小、类型限制, 或上传路径不存在
    e.printStackTrace(); //此处可根据需要做适当的异常处理, 如弹窗提示用户
}
```

如果是同时上传多个文件, 可以使用 for 循环遍历各文件, 使用 su.getFiles().getCount() 可以获取上传的文件个数, 以此作为循环条件。在处理上传文件发生的异常时, 经过测试, 异常信息中包含了四位整数代表的错误代号, 其中 1101 表示上传类型不合法, 1105 表示超出了单个上传文件的大小限制, 1135 表示上传路径不存在, 用户可以使用 indexOf() 方法来判断异常文本中是否包含以上错误代号, 从而给出不同的用户提示, 提高用户界面的友好性。

2 Apache Commons FileUpload 组件

Commons FileUpload 是 Apache 组织免费提供的文件上传组件, 可以直接从该组件官网下载, 但是 Commons FileUpload 组件还依赖 Commons IO 组件, 因此还需要下载 Commons

收稿日期: 2016-03-10

作者简介: 陈涛(1979—), 男, 河南商丘人, 副教授, 硕士, 研究方向为计算机网络; 黄艳峰(1977—), 女, 河南兰考人, 副教授, 硕士, 研究方向为计算机软件与理论。

IO 组件包。该组件使用起来也很方便,它提供了更为高效、安全的文件上传方式,目前也是很多 Java Web 框架(如 Struts、Spring 等)所支持的文件上传组件。

Commons FileUpload 的 API 中主要包括 DiskFileItemFactory、ServletFileUpload、FileItem 等核心类。首先需要把关于 Commons FileUpload 和 Commons IO 的两个 jar 包拷贝至 WEB-INF\lib 目录下,然后编写实现上传文件功能的 JSP 文件。文件上传表单的代码与上面相同。使用 Commons FileUpload 上传文件的核心代码如下:

```
DiskFileItemFactory factory = new DiskFileItemFactory(); //
创建磁盘文件工厂对象
```

```
ServletFileUpload upload = new ServletFileUpload(factory); //
创建文件上传对象
```

```
upload.setFileSizeMax(300*1024*1024); //设置单个文件最
大上传大小,如 300M
```

```
try {
    List<FileItem> items = upload.parseRequest(request); //解析
请求发来的信息
```

```
    Iterator<FileItem> fileItemsIterator = fileItemsList.iterator();
//遍历处理上传的 FileItem
```

```
    while(fileItemsIterator.hasNext()) {
        FileItem fileItem = fileItemsIterator.next();
        if(fileItem.isFormField()) { //如果 FileItem 是非文件类
型的普通表单字段
```

```
            System.out.println(fileItem.getString()); //此处打印
字段值,也可做其它处理
```

```
        } else {
            String filename = fileItem.getName();
            filename=filename.substring(filename.lastIndexOf("\\")+1); //
从路径提出文件名
```

```
            String fileExt = filename.substring(filename.lastIndexOf(".")
+1); //得到文件扩展名
```

```
            File savePath = request.getServletContext().getRealPath("up-
load\\")+filename;
```

```
            fileItem.write(savePath); //保存上传的文件
```

```
        }
    } catch (FileUploadException e) {
        e.printStackTrace();
    }
}
```

注意不同浏览器的默认设置不同,IE 浏览器默认提交文件的整个路径(可以在 Internet 安全设置更改为不包含本地路径),而火狐或谷歌浏览器默认只提交文件名,所以需要考虑可能包含路径的情况,把文件名从中提取出来。Commons FileUpload 组件自身没有提供限制上传文件类型的方法,需要自行编码从文件名取出扩展名并与允许上传的文件扩展名进行比较,然后给出用户相应的错误提示。除了使用 FileItem 的 write() 方法保存文件外,还可以通过 FileItem 的 getInputStream() 方法获取文件输入流,然后再通过输出流保存至文件甚至是数据库。在上传文件时如果超出了文件大小限制,系统会抛出异常 FileSizeLimitExceedException,此异常是 FileUploadException 异常的子类,所以可以捕获该异常,并在异常处理代码中给出用户错误提示。

3 Servlet 3.0 文件上传

上述两种文件上传方法都需要用到第三方组件,而在最新的 Servlet 3.0 规范中,提供了对文件上传的原生支持,直接使用 Servlet 3.0 提供的 API 就能够轻松实现文件上传。较之 Servlet 2.5 主要是增加了两个类: MultipartConfig 和 Part。其中 MultipartConfig 是注解类,要想让 Servlet 支持文件上传只需在其前面添加如下形式注解即可:

```
@MultipartConfig(maxFileSize=1024*1024*300) //设置单个
文件最大上传大小,如 300M
```

另外 HttpServletRequest 类还增加了一个 getParts() 方法,该方法可以获取表单提交过来的所有上传文件,这些文件信息封装在一个个的 Part 对象中。Part 类提供了 write 方法用于保存上传的文件。在 Servlet 的 doPost() 方法编写如下代码:

```
for(Part part : request.getParts()) { //遍历所有上传的文件
    String filename = getFilename(part); //自定义方法从请求头
信息中得到文件名
```

```
    File savePath = request.getServletContext().getRealPath("up-
load\\")+filename;
```

```
    part.write(savePath); //保存上传的文件到指定的路径
}
```

因为 Servlet 3.0 没有提供直接获取文件名的方法,所以只能自行编写方法 getFilename() 从请求头中解析文件名。以火狐浏览器为例(若是 IE,则文件名带路径),请求头格式为:

```
Content-Disposition: form-data; name="fileName"; filename=
"test.avi"。getFilename() 方法代码如下:
```

```
private String getFilename(Part part) {
    String contentDisp = part.getHeader("content-disposition");
    String[] tokens = contentDisp.split(";");
    for (String token : tokens) {
        if (token.trim().startsWith("filename=")) {
            String filename = token.substring(token.indexOf("=") +
2, token.length()-1);
            return filename.substring(filename.lastIndexOf("\\")+1);
//兼容所有浏览器
        }
    }
    return "";
}
```

如果上传文件违法大小限制,在获取 Part 对象时将抛出 IllegalStateException 异常,所以可以将以上 for 循环代码置入 try-catch 结构中,通过捕获 IllegalStateException 异常来判断文件是否超出大小限制。对于上传文件后缀的限制方法与 Commons FileUpload 组件一样,也是在得到的文件名中提取扩展名,再与允许上传的文件扩展名比较即可。

4 运行测试

分别运行以上三种文件上传程序,当上传体积较小文件时,三者资源利用率及响应时间方面差别不太显著,但当增加上传文件大小时,例如上传 300M 左右的视频文件,经测试使用 SmartUpload 组件、Commons FileUpload 组件、Servlet 3.0 API 上传文件花费的平均时间分别为 6.642s、5.174s、4.750s。从资源占用情况来看,Servlet 3.0 API 文件上传也是最节约系统资源的方法。(下转第 52 页)

```

Password);
    smtp.Port = 25;
    smtp.Host = "smtp.qq.com";
    smtp.EnableSsl = false;
    try{
        smtp.Send(myEmail);
    }
    catch (Exception ex){
        throw new Exception(ex.Message);
    }
}

```

5 数据访问

通过数据库对数据进行管理,能够非常便利地实现相关数据的查询与更新。系统主要采用SQL Server2008作为数据库管理工具对系统中相关的数据表结构进行设计^[9]。

1) 用户信息表:该表主要存放系统中用户的基本信息,其中用户包含各级管理员。所包含的主要字段有:用户名、密码、姓名、身份证号、职称、学历、邮箱、电话、单位等。

2) 项目信息表:该表主要存放系统中项目信息,所包含的主要字段有:项目类别、项目来源、项目名称、项目合同、开始时间、结束时间、主持单位、主持人、项目经费等。

3) 项目成员信息表:该表主要存放项目成员信息,主要字段为:项目ID、成员姓名、身份证号、学历、职称、年工作月数、成员排序、工作单位、主要工作内容等。

4) 项目预算表:该表主要存放项目经费预算信息,主要字

段有:设备费,设备费内容、材料费、材料费内容、测试化验费、测试化验费内容、燃料动力费、燃料动力费内容、差旅费、差旅费内容、会议费、会议费内容等。

不仅如此,在数据库中还包含了所在单位、项目类别、项目评审表、用户登录信息表等其他相关的数据表。所在单位表记录单位信息,项目类别表记录项目类别信息、项目评审表记录项目评审信息包含评审意见、评审结果、评审时间、评审人等,用户登录信息表记录用户登录时间、登录人、登录IP地址等。

6 结束语

本文采用B/S模式,利用Visual Studio 2010程序开发工具,以C#为编程语言,ASP.NET 三层架构,SQL Server2008数据库,实现了科技信息管理系统的设计。科技信息管理系统的使用能够大大减少各级科研管理人员的繁杂劳动,提高科研管理效率和水平,为科研管理和决策提供了重要依据,具有重要的应用价值。

参考文献:

- [1] 周金桥. ASP.NET 夜话[M]. 北京: 电子工业出版社, 2009.
- [2] PageOffice 插件官网[EB/OL]. <http://www.zhuzhengsoft.com/>.
- [3] 晓红, 谢素萍. 高校科研管理信息系统构建与完善路径研究[J]. 高等工程教育研究, 2015(1): 107-112.
- [4] 许博. 基于.NET 三层架构的科研管理系统的开发与应用[J]. 实验室研究与探索, 2013(2): 188-190.
- [5] 丁玲. ASP.NET 环境下的院校人才就业平台的设计与实现[J]. 电脑知识与技术, 2016(1): 68-69.

(上接第49页)

5 结论

本文分析对比了目前Java Web开发中三种常见的文件上传方法,前两种方法需要借助第三方组件来实现,其中SmartUpload组件代码简单,实现方便,但目前已停止更新,并且在上传大文件时效率不高,而Commons FileUpload组件目前依然在更新维护中,上传大文件时有不错的性能,并且也被很多主流框架所使用。最后一种文件上传方法使用Servlet 3.0自身的API来实现,不再依赖第三方组件,上传效率比Commons FileU-

pload也更为高效,所以如果Java Web项目是基于Java EE 6以上版本开发的,建议考虑使用Servlet 3.0 API的文件上传方法。

参考文献:

- [1] 张书峰. 基于Java语言的文件上传组件研究[J]. 电脑知识与技术, 2013(33).
- [2] 刘杨. JSP项目开发常用文件上传组件比较及举例[J]. 电脑编程技巧及维护, 2015(8).
- [3] 周兵. 基于JSP文件上传中的问题探讨及改进方案[J]. 兰州工业学院学报, 2015(4).