今日内容

文件的上传和下载

文件的上传和下载，是非常常见的功能。很多的系统中，或者软件中都经常使用文件的上传和下载。 比如：QQ 头像，就使用了上传。 邮箱中也有附件的上传和下载功能。 OA 系统中审批有附件材料的上传。

1、文件的上传介绍（*重点）

1、要有一个 form 标签，method=post 请求
2、form 标签的 encType 属性值必须为 multipart/form-data 值
3、在 form 标签中使用 input type=file 添加上传的文件
4、编写服务器代码（Servlet 程序）接收，处理上传的数据。

encType=multipart/form-data 表示提交的数据，以多段（每一个表单项一个数据段）的形式进行拼接，然后以二进制流的形式发送给服务器

1.1、 文件上传，HTTP 协议的说明。

1.2、commons-fileupload.jar 常用 API 介绍说明

commons-fileupload.jar 需要依赖 commons-io.jar 这个包，所以两个包我们都要引入。

第一步，就是需要导入两个 jar 包：

- commons-fileupload-1.2.1.jar
- commons-io-1.4.jar

commons-fileupload.jar 和 commons-io.jar 包中，我们常用的类有哪些?

- ServletFileUpload 类，用于解析上传的数据。
- FileItem 类，表示每一个表单项。

- boolean ServletFileUpload.*isMultipartContent*(HttpServletRequest request);

> 判断当前上传的数据格式是否是多段的格式。

- public List parseRequest(HttpServletRequest request)

> 解析上传的数据

- boolean FileItem.isFormField()

> 判断当前这个表单项，是否是普通的表单项。还是上传的文件类型。
> true 表示普通类型的表单项
> false 表示上传的文件类型

- String FileItem.getFieldName()

> 获取表单项的 name 属性值

- String FileItem.getString()

> 获取当前表单项的值。

- String FileItem.getName();

> 获取上传的文件名

- void FileItem.write(file);

> 将上传的文件写到 参数 file 所指向抽硬盘位置 。

1.3、fileupload 类库的使用

上传文件的表单：

```
<%--
  Created by IntelliJ IDEA.
  User: baishixin
  Date: 2020/9/1
  Time: 1:02 上午
  To change this template use File | Settings | File Templates.
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
```

```
    <title>Title</title>
</head>
<body>
    <form action="http://localhost:8080/05_JSP/upload" method="post"
enctype="multipart/form-data">
        账号:<input type="text" name="username"><br>
        头像:<input type="file" name="photo"><br>
        <input type="submit" value="上传">
    </form>
</body>
</html>
```

解析上传的数据的代码:

```
package com.admiral.web;

import org.apache.commons.fileupload.FileItem;
import org.apache.commons.fileupload.FileItemFactory;
import org.apache.commons.fileupload.FileUploadException;
import org.apache.commons.fileupload.disk.DiskFileItemFactory;
import org.apache.commons.fileupload.servlet.ServletFileUpload;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.File;
import java.io.IOException;
import java.util.List;

public class UploadServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {

        //1.判断上传的数据是否为多段数据
        if(ServletFileUpload.isMultipartContent(request)){
            FileItemFactory fileItemFactory = new DiskFileItemFactory();
            ServletFileUpload servletFileUpload = new
ServletFileUpload(fileItemFactory);

            try {
                List<FileItem> list = servletFileUpload.parseRequest(request);

                for (FileItem fileItem : list) {
                    if(fileItem.isFormField()){
                        //普通表单项
                        System.out.println(fileItem.getFieldName());
                        System.out.println(fileItem.getString());
```

```
                    }else {
                        System.out.println(fileItem.getFieldName());
                        System.out.println(fileItem.getName());
                        fileItem.write(new File("/Users/baishixin/Downloads/"
+ fileItem.getName() ));
                    }
                }

            } catch (Exception e) {
                e.printStackTrace();
            }

        }else {

        }

    }

}
```

2、文件下载

下载的常用 API 说明：

- response.getOutputStream();
- servletContext.getResourceAsStream();
- servletContext.getMimeType();
- response.setContentType();
- response.setHeader("Content-Disposition", "attachment; fileName=1.jpg");

这个响应头告诉浏览器。这是需要下载的。而 attachment 表示附件，也就是下载的一个文件。fileName=后面， 表示下载的文件名。

完成上面的两个步骤，下载文件是没问题了。但是如果我们要下载的文件是中文名的话。你会发现，下载无法正确显示出正确的中文名。

原因是在响应头中，不能包含有中文字符，只能包含 ASCII 码。

文件下载实例代码:

```
package com.qidi.servlet;

import org.apache.commons.io.IOUtils;
```

```
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.ServletOutputStream;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;

public class DownloadServlet extends HttpServlet {

    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        //1.获取要下载的文件
        String downFileName = "1.jpeg";

        //2.读取要下载的文件
        ServletContext servletContext = getServletContext();

        //4.设置要下载文件的类型
        String mimeType = servletContext.getMimeType("/file/" + downFileName);
        response.setContentType(mimeType);

        //5.要告诉客户端文件是用来下载的
        response.setHeader("content-Disposition","attachment;
filename=22222.jpeg");

        InputStream resourceAsStream =
servletContext.getResourceAsStream("/file/" + downFileName);


        //3.回传数据给客户端
        OutputStream outputStream = response.getOutputStream();
        System.out.println(resourceAsStream);
        System.out.println(outputStream);
        IOUtils.copy(resourceAsStream,outputStream);

    }
}
```

附件中文名乱码问题解决方案

方案一：URLEncoder 解决 IE 和谷歌浏览器的 附件中 文名问题。

如果客户端浏览器是 IE 浏览器 或者 是谷歌浏览器。我们需要使用 URLEncoder 类先对中文名进行 UTF-8 的编码操作。

因为 IE 浏览器和谷歌浏览器收到含有编码后的字符串后会以 UTF-8 字符集进行解码显示。

```
// 把中文名进行 UTF-8 编码操作。
String str = "attachment; fileName=" + URLEncoder.encode("中文.jpg", "UTF-8");
// 然后把编码后的字符串设置到响应头中
response.setHeader("Content-Disposition", str);
```

方案二：BASE64 编解码 解决 火狐浏览器的附件中文名问 题

如果客户端浏览器是火狐浏览器。 那么我们需要对中文名进行 BASE64 的编码操作。

```
这时候需要把请求头
Content-Disposition: attachment; filename=中文名
编码成为：
Content-Disposition: attachment; filename==?charset?B?xxxxx?=
```

因为火狐使用的是 BASE64 的编解码方式还原响应中的汉字。所以需要使用 BASE64Encoder 类进行编码操作。

```
// 使用下面的格式进行 BASE64 编码后
String str = "attachment; fileName=" + "=?utf-8?B?" + new
BASE64Encoder().encode("中文.jpg".getBytes("utf-8")) + "?=";
// 设置到响应头中
response.setHeader("Content-Disposition", str);
```

那么我们如何解决上面两种不同编解码方式呢。我们只需要通过判断请求头中 User-Agent 这个请求头携带过来的

浏览器信息即可判断出是什么浏览器。

```
String ua = request.getHeader("User-Agent");

// 判断是否是火狐浏览器

  if(ua.contains("Firefox")) {

        // 使用下面的格式进行 BASE64 编码后
        String str = "attachment; fileName=" + "=?utf-8?B?" \+ **new**
BASE64Encoder().encode("中文.jpg".getBytes("utf-8")) + "?=";

        // 设置到响应头中
        response.setHeader("Content-Disposition", str);
```

```
  } else {

        // 把中文名进行 UTF-8 编码操作。
        String str = "attachment; fileName=" + URLEncoder.*encode*("中文.jpg",
"UTF-8");

        // 然后把编码后的字符串设置到响应头中
        response.setHeader("Content-Disposition", str);

}
```