

14.上传下载

CONTENTS

● 技能 上传下载

● 练习

练习1——理解上传下载的基本思路

练习2——开始上传并理解上传API的作用

练习3——实现上传文件到指定目录

练习4——完善上传文件到指定目录

练习5——保存上传信息到数据库

练习6——将服务器的图片在客户端下载

练习7——完善将服务器的图片在客户端下载



练习1——理解上传下载的基本思路

- 需求说明：
 - 演示最终上传下载效果
 - 提供上传下载素材
- 理解上传思路：
 - 稍微复杂一些 直接使用Servlet实现有难度，一般使用apache commons组件中commons-fileUpload组件，大大降低操作的难度。
 - commons-fileUpload离不开commons-io组件
 - commons-fileUpload的常用类
 - FileItemFactory DiskFileItemFactory：工厂类，生产FileItem
 - FileItem：每个表单项都会被封装成一个FileItem对象
 - ServletFileUpload：实现上传操作，将表单数据封装到FileItem中
- 理解下载思路：
 - 不需要额外组件，会使用IO流，是连接服务器和客户端，不是本地的
 - 使用response内建对象得到发送给客户端的输出流



练习2——开始上传并理解上传API的作用

- 需求说明：
 - 上传文件到服务器，不保存相关信息到数据库，只涉及视图层和控制层
- 实现步骤
 - 添加jar
 - commons-fileupload.jar commons-io.jar
- 视图层：
 - method="post"
 - enctype="multipart/form-data"
 - 照片:<input type="file" name="photo">

- 控制层：
 - 掌握基本步骤：4步
 - 重点掌握FileItem的主要方法



练习2——开始上传并理解上传API的作用

- FileItem的常用方法

- item.isFormField() 如果是普通表单项 true, 如果是file表单项, false
- item.getFieldName() 表单项的name属性
- item.getString() 普通表单项的值, file表单项就是file的内容
- item.getString("utf-8") 解决普通表单项的中文乱码问题
- item.getName() 普通表单项就是null, file表单项就是文件的名称
- item.getContentType() 普通表单项是null, file表单项就是文件的类型
- item.getSize(): 表单项内容长度

- 掌握如何解决中文乱码问题

- 解决file表单项中文乱码问题:
 - upload.setHeaderEncoding("utf-8");
- 解决普通表单项的中文乱码问题:
 - item.getString("utf-8")



练习3——上传文件到指定目录

- 需求说明：
 - 上传文件到服务器，不保存相关信息到数据库，只涉及视图层和控制层

- 注意事项

- 需要对普通表单项和file表单项分别进行处理

```
if(!item.isFormField()){//文件表单项
    if("photo".equals(item.getFieldName())){
        //获取文件的名称
        String fileName = item.getName();
        //指定要上传的位置
        File file = new File("d:/upload/"+fileName);
        //上传文件到服务器指定位置
        item.write(file);
    }
}
```

- 目录必须已经存在，否则提示路径找不到，需要动态创建



练习4——完善上传文件到指定目录

- 完善1：限制上传文件的大小
 - 方式1： `upload.setSizeMax(1024);` 推荐使用
 - 方式2： `long size = item.getSize();` 不推荐使用
- 完善2：限制上传文件的类型
 - `String fileName = item.getName();` // .jpg JPG adf.ad.jpg adfad.gif
 - `String extName = fileName.substring(fileName.lastIndexOf(".") + 1).toLowerCase();`
- 完善3：为了避免同名文件的覆盖，文件重命名
 - `long newFileName = UUID.randomUUID();`
 - `fileName = newFileName + "." + extName;`



练习5——保存上传信息到数据库

- 需求说明：
 - 保存上传文件相关信息到数据库
 - 根据数据库中存储的路径显示图片
- 实现步骤
 - 数据库表
 - 数据库表中同时保持真实名称和防止重复的名称
 - STUDENT增加三个字段：photoName, realName, photoType
 - 实体类
 - Student类中增加三个属性
 - 控制层:无难度
 - 业务层:无难度
 - DAO层:无难度



练习6——将服务器的图片在客户端下载

- 需求说明：
 - 将服务器的图片在客户端下载
- 关键代码

```
//创建输入流和输出流
String photoName = stu.getPhotoName();
String realPath = this.getServletContext().getRealPath("/upload");
//InputStream is = new FileInputStream("D:/Java/apache-tomcat-
//7.0.69/webapps/updownload0/upload/"+photoName);
InputStream is = new FileInputStream(realPath+"/"+photoName);
OutputStream os = response.getOutputStream();//发送到客户端
//3.2使用输入流和输出流完成文件传递
IOUtils.copy(is, os);
//3.3关闭输入流和输出流
is.close();
os.close();
```



练习7——完善将服务器图片在客户端下载

- 问题1：下载的文件是直接显示，而不是提示下载或者另存为

```
response.setContentLength((int)length);  
response.setContentType(photoType);  
//response.setHeader("Content-Disposition", "inline");  
response.setHeader("Content-Disposition", "attachment;filename="+stu.getPhotoName());//附件
```

- 问题2：如果文件名中有中文，会出现乱码问题，进而影响下载

```
String userAgent = request.getHeader("User-Agent").toLowerCase();  
if(userAgent.indexOf("msie")>=0){ //msie  
    photoName2 = URLEncoder.encode(photoName, "utf-8");  
}else{  
    byte [] bytes = photoName.getBytes("utf-8");  
    photoName2 = new String(bytes,"iso-8859-1");  
}
```



练习7——完善将服务器图片在客户端下载

- 问题3：把上传的文件保存在非服务器路径下d:\upload和保存在服务器目录路径下\updownload\upload目录下有没有区别
 - 情况1：服务器路径下\updownload\upload
 - 内容可以直接访问
 - ``
 - 注意：删除项目时需要先备份上传的内容，否则丢失了
 - 情况2：d:\upload
 - 用户通过浏览器访问tomcat，tomcat的根目录是webapps，所以d:\upload下内容不能直接访问
 - 删除项目时，上传的内容因为不在项目目录中，所以不受影响