

# SpringMVC文件上传下载

## 一、文件上传

### 1、控制层核心步骤

- 1.获取文件名
- 2.对文件进行处理比如(生成随机文件名)
  - a.获取文件后缀名
  - b.生成随机文件名
- 3.获取存放文件的物理路径
- 4.创建文件实例
- 5、写入文件

### 2、上传代码实现

#### A、添加依赖

```
1 <dependencies>
2   <!--加载spring依赖 -->
3   <dependency>
4     <groupId>org.springframework</groupId>
5     <artifactId>spring-webmvc</artifactId>
6     <version>5.2.4.RELEASE</version>
7   </dependency>
8   <dependency>
9     <groupId>commons-fileupload</groupId>
10    <artifactId>commons-fileupload</artifactId>
11    <version>1.3.2</version>
12  </dependency>
13 </dependencies>
```

#### B、配置文件编写

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xmlns:p="http://www.springframework.org/schema/p"
5   xmlns:mvc="http://www.springframework.org/schema/mvc"
6   xmlns:context="http://www.springframework.org/schema/context"
7   xsi:schemaLocation="http://www.springframework.org/schema/mvc
8     http://www.springframework.org/schema/mvc/spring-mvc.xsd
9     http://www.springframework.org/schema/context
```

```

10      http://www.springframework.org/schema/context/spring-context.xsd
11      http://www.springframework.org/schema/beans
12      http://www.springframework.org/schema/beans/spring-beans.xsd">
13      <!--扫描组件 -->
14      <!--告诉Spring 到哪里去找标记为@Controller 的Controller控制器-->
15      <context:component-scan base-package="com.sm.controller"/>
16      <!--启用注解驱动-mvc配置 -->
17      <mvc:annotation-driven/>
18      <!--配置视图解析器 -->
19      <bean id="viewResolver"
20      class="org.springframework.web.servlet.view.InternalResourceViewResolver"
21      p:prefix="/WEB-INF/jsp/" p:suffix=".jsp"/>
22
23      <!--添加multipartResolver的bean支持文件上传
24      maxUploadSize 上传文件的最大大小，单位为字节
25      -->
26      <bean id="multipartResolver"
27      class="org.springframework.web.multipart.commons.CommonsMultipartResolver"
28      p:maxUploadSize="802400" p:defaultEncoding="utf-8"/>
29  </beans>

```

## C、添加表单

```

1  <form action="doUpload.do" method="post" enctype="multipart/form-data">
2      图片:<input type="file" name="file"/><input type="submit"/>
3  </form>

```

## D、控制层代码

```

1  @RequestMapping("/doUpload.do")
2  public ModelAndView toUpload(MultipartFile file,HttpServletRequest request)
3  throws Exception{
4      //1.获取文件名
5      String picName=file.getOriginalFilename();
6      //2.对文件进行处理比如(生成随机文件名)
7      //a.获取文件后缀名
8      String ext=FilenameUtils.getExtension(picName);
9      //b.生成随机文件名
10     String savePicName=System.currentTimeMillis()+ "."+ext;
11     //3.获取存放文件的物理路径
12     String savePath="D:/kclz/ws/mslx/05-SpringMVC_Demo03/webContent/upload";
13     System.out.println(savePath);
14     //4.创建文件实例
15     File fl=new File(savePath,savePicName);
16     //5、写入文件
17     Fileutils.copyInputStreamToFile(file.getInputStream(), fl);
18     //file.transferTo(fl);
19     return new ModelAndView("index");
20 }

```

## E、测试

```
1 | localhost:8080/05-SpringMVC_Demo03/toIndex.do
```

## 3、注意事项

### A、表单上

表单上要添加 `enctype="multipart/form-data"`，且input的type类型为file

### B、配置文件上

```
1 | <!--添加multipartResolver的bean支持文件上传 maxUploadSize
2 | 上传文件的最大大小，单位为字节-->
3 | <bean id="multipartResolver"
4 | class="org.springframework.web.multipart.commons.CommonsMultipartResolver"
   | p:maxUploadSize="802400" p:defaultEncoding="utf-8"/>
```

### C、控制层方法上面

控制层方法参数添加MultipartFile file和HttpServletRequest request

且file这个变量名要和表单的name值一致

## 二、文件下载

### 1、控制层核心步骤

- A、获得文件路径
- B、构建一个输入流对象
- C、中文乱码处理
- D、通过response设置输出格式和内容
- E、构建一个缓冲区
- F、循环读写
- G、刷新和关闭资源

### 2、下载实现

#### A、下载链接

```
1 | <h2><a href="download.do?filename=飞机.jpg">点击下载</a></h2>
```

#### B、控制层代码

```
1 | @RequestMapping("/download.do")
```

```

2 public void downLoad(String filename,HttpServletRequest req,HttpServletResponse res)
  throws Exception{
3     System.out.println(filename);
4     //1、获得文件
5     //String fname=req.getServletContext().getRealPath("/upload/"+filename);
6     String fname="D:/kclz/ws/mslx/05-SpringMVC_Demo03/WebContent/upload/"+filename;
7     //System.out.println(fname);
8     //2、通过一个文件构建一个输入流对象
9     FileInputStream fis=new FileInputStream(fname);
10    //3、中文乱码的问题(默认处理字符根据操作系统GBK-但是浏览器接收这个值的时候 采用的是iso-8859-1
    来解析)
11    filename=new String(filename.getBytes("GBK"),"ISO-8859-1");
12    //3、通过response设置输出格式和内容
13    res.setHeader("content-disposition","attchment;filename="+filename);
14    res.setContentType("application/octet-stream");//文件类型
15    res.setContentLength(fis.available());//设置文件大小
16    //4、构建一个缓冲区
17    byte[] by=new byte[2048];//构建一个缓冲区
18    //5、循环读写
19    int len=0;
20    while((len=fis.read(by))!=-1){
21        res.getOutputStream().write(by,0,len);
22    }
23    //6、刷新
24    res.getOutputStream().flush();
25    //7、关闭资源
26    res.getOutputStream().close();
27    fis.close();
28 }

```

### 3、注意事项

```

1 //下载中文文件名文件记得处理乱码
2 filename=new String(filename.getBytes("GBK"),"ISO-8859-1");

```

## 三、更多视频扩展资源

今日头条：<https://www.toutiao.com/c/user/1736832959129566/> 进去后-选择视频-在线点击观看

哔哩哔哩：<https://space.bilibili.com/425233867>