文件上传

1. 加入jar包



1. 配置SpringMVC文件

配置一个*CommonsMultipartResolver*对象，SpringMVC并未提供任何上传组件，所以这个对象来源于阿帕奇的jar包中，但SpringMVC提供了对对象的使用，使用的是MultipartFile接口

<!-- 配置文件上传对象 -->

<bean id=*"multipartResolver"* class=*"org.springframework.web.multipart.commons.***CommonsMultipartResolver***"*>

<!-- 配置几个属性，默认字符编码 -->

<property name=*"defaultEncoding"* value=*"UTF-8"*></property>

<!-- 最大上传大小 -->

<property name=*"maxUploadSize"* value=*"1024000"*></property>

</bean>

1. 写一个表单域

<form action=*"testFileLoad"* method=*"post"* enctype=*"multipart/form-data"*>

File：<input type=*"file"* name=*"file"*/><br>

Desc：<input type=*"text"* name=*"desc"*><br>

<input type=*"submit"* value=*"Submit"* />

</form>

1. 写一个响应函数

表单域中文件的对象被自动转化为了一个MultipartFile对象，可以用@RequestParam注解直接获取，可以获取文件的对应属性和文件的输入流

@RequestMapping("/testFileLoad")

**public** String testFileLoad(@RequestParam("desc") String desc, **@RequestParam("file") MultipartFile file**){

System.***out***.println("desc：" + desc);

// 打印原始文件名

System.***out***.println("OriginalFilename：" + file.getOriginalFilename());

//获取上传文件的输入流

InputStream in=file.getInputStream();

//创建一个指定文件夹的输出流

OutputStream os = **new** FileOutputStream(**new** File("E:\\file\\" + file.getOriginalFilename()));

//上传

**byte** [] bs=**new** **byte**[1024];

**int** len=0;

**while**((len=in.read(bs))!=-1){

os.write(bs, 0, len);

}

os.close();

in.close();

**return** "success";

}

https://blog.csdn.net/u014746965/article/details/78772896