1. 文件路径的斜杠“//”用File.separator代替.
2. 检查文件存在：File.exists(Paths.get(savepath+File.separator+fileName)).
3. 文件上传：

前端：用jQuery插件jQuery.form.js和ajaxsubmit（）方法，type属性要进行设置，一般为post。

后台：Controller层 public jsonResult uploadFile(MultipartFile file){} file为前端上传文件输入框的name=“file”。

Service层处理文件有两种方法，

第一种：service.uploadFile(MultipartFile file){

File oFile=new File(savepath,file.getOriginalFilename());

File bFile=new File(savepath, file.getOriginalFilename()+”.pdf”);

If(file!=null){

//如果需要文件覆盖

deleteFile(bFile);

if(oFile.isFile()){

//将ofile重命名为bFile

oFile.renameTo(bFile);

}

saveFile(file,fileName,savepath)

}

}。

Public void saveFile(MultipartFile file,String filename,String savepath){

File targetFile=new File(savepath,filename);

File.transferTo(targetFile);

}

第二种:service.uploadFile(InputStream file.getInputStream){

//断言

Assert.notNull(“savepath”,”路径不能为空”);

Files.copy(FileInputStream,Paths.get(savepath,filename,true));

}

1. 多文件上传：
2. @Controller
3. @RequestMapping("/file")
4. **public** **class** UploadController {

7. @RequestMapping("/upload"   )
8. **public** String addUser(@RequestParam("file") CommonsMultipartFile[] files,HttpServletRequest request){
10. **for**(**int** i = 0;i<files.length;i++){
11. System.out.println("fileName---------->" + files[i].getOriginalFilename());
13. **if**(!files[i].isEmpty()){
14. **int** pre = (**int**) System.currentTimeMillis();
15. **try** {
16. //拿到输出流，同时重命名上传的文件
17. FileOutputStream os = **new** FileOutputStream("H:/" + **new** Date().getTime() + files[i].getOriginalFilename());
18. //拿到上传文件的输入流
19. FileInputStream in = (FileInputStream) files[i].getInputStream();
21. //以写字节的方式写文件
22. **int** b = 0;
23. **while**((b=in.read()) != -1){
24. os.write(b);
25. }
26. os.flush();
27. os.close();
28. in.close();
29. **int** finaltime = (**int**) System.currentTimeMillis();
30. System.out.println(finaltime - pre);
32. } **catch** (Exception e) {
33. e.printStackTrace();
34. System.out.println("上传出错");
35. }
36. }
37. }
38. **return** "/success";
39. }

42. @RequestMapping("/upload2"  )
43. **public** String upload2(HttpServletRequest request,HttpServletResponse response) **throws** IllegalStateException, IOException {
44. //创建一个通用的多部分解析器
45. CommonsMultipartResolver multipartResolver = **new** CommonsMultipartResolver(request.getSession().getServletContext());
46. //判断 request 是否有文件上传,即多部分请求
47. **if**(multipartResolver.isMultipart(request)){
48. //转换成多部分request
49. MultipartHttpServletRequest multiRequest = (MultipartHttpServletRequest)request;
50. //取得request中的所有文件名
51. Iterator<String> iter = multiRequest.getFileNames();
52. **while**(iter.hasNext()){
53. //记录上传过程起始时的时间，用来计算上传时间
54. **int** pre = (**int**) System.currentTimeMillis();
55. //取得上传文件
56. MultipartFile file = multiRequest.getFile(iter.next());
57. **if**(file != **null**){
58. //取得当前上传文件的文件名称
59. String myFileName = file.getOriginalFilename();
60. //如果名称不为“”,说明该文件存在，否则说明该文件不存在
61. **if**(myFileName.trim() !=""){
62. System.out.println(myFileName);
63. //重命名上传后的文件名
64. String fileName = "demoUpload" + file.getOriginalFilename();
65. //定义上传路径
66. String path = "H:/" + fileName;
67. File localFile = **new** File(path);
68. file.transferTo(localFile);
69. }
70. }
71. //记录上传该文件后的时间
72. **int** finaltime = (**int**) System.currentTimeMillis();
73. System.out.println(finaltime - pre);
74. }
76. }
77. **return** "/success";
78. }
80. @RequestMapping("/toUpload" )
81. **public** String toUpload() {
83. **return** "/upload";
84. }
86. }