[SpringMVC 文件上传配置，多文件上传，使用的MultipartFile](http://blog.csdn.net/swingpyzf/article/details/20230865)

标签： [文件上传](http://www.csdn.net/tag/%e6%96%87%e4%bb%b6%e4%b8%8a%e4%bc%a0)[SpringMVC](http://www.csdn.net/tag/SpringMVC)[MultipartFile](http://www.csdn.net/tag/MultipartFile)

2014-03-01 20:17 67436人阅读 [评论](http://blog.csdn.net/swingpyzf/article/details/20230865#comments)(21) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/swingpyzf/article/details/20230865#report)

http://static.blog.csdn.net/images/category_icon.jpg 分类：

Java（22） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg Spring框架（3） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg

版权声明：本文为博主原创文章，未经博主允许不得转载。

**基本的SpringMVC的搭建在我的上一篇文章里已经写过了，这篇文章主要说明一下如何使用SpringMVC进行表单上的文件上传以及多个文件同时上传的步骤  
  
SpringMVC 基础教程 框架分析：**[**http://blog.csdn.net/swingpyzf/article/details/8885459**](http://blog.csdn.net/swingpyzf/article/details/8885459)

**SpringMVC 基础教程 简单入门实例：**[**http://blog.csdn.net/swingpyzf/article/details/8904205**](http://blog.csdn.net/swingpyzf/article/details/8904205)

**文件上传项目的源码下载地址：[http://download.csdn.net/detail/swingpyzf/6979915](http://download.csdn.net/detail/swingpyzf/6979915" \t "_blank)**

一、配置文件：  
SpringMVC 用的是 的MultipartFile来进行文件上传 所以我们首先要配置MultipartResolver:用于处理表单中的file

**[html]** [view plain](http://blog.csdn.net/swingpyzf/article/details/20230865) [copy](http://blog.csdn.net/swingpyzf/article/details/20230865)

[print](http://blog.csdn.net/swingpyzf/article/details/20230865)[?](http://blog.csdn.net/swingpyzf/article/details/20230865)[在CODE上查看代码片](https://code.csdn.net/snippets/212406)

1. <!-- 配置MultipartResolver 用于文件上传 使用spring的CommosMultipartResolver -->
2. **<beans:bean** id="multipartResolver" class="org.springframework.web.multipart.commons.CommonsMultipartResolver"
3. p:defaultEncoding="UTF-8"
4. p:maxUploadSize="5400000"
5. p:uploadTempDir="fileUpload/temp"
6. **>**
7. **</beans:bean>**

其中属性详解：  
defaultEncoding="UTF-8" 是请求的编码格式，默认为iso-8859-1  
maxUploadSize="5400000" 是上传文件的大小，单位为字节  
uploadTempDir="fileUpload/temp" 为上传文件的临时路径

二、创建一个简单的上传表单：

**[html]** [view plain](http://blog.csdn.net/swingpyzf/article/details/20230865) [copy](http://blog.csdn.net/swingpyzf/article/details/20230865)

[print](http://blog.csdn.net/swingpyzf/article/details/20230865)[?](http://blog.csdn.net/swingpyzf/article/details/20230865)[在CODE上查看代码片](https://code.csdn.net/snippets/212406)

1. **<body>**
2. **<h2>**文件上传实例**</h2>**

5. **<form** action="fileUpload.html" method="post" enctype="multipart/form-data"**>**
6. 选择文件:**<input** type="file" name="file"**>**
7. **<input** type="submit" value="提交"**>**
8. **</form>**

11. **</body>**

注意要在form标签中加上enctype="multipart/form-data"表示该表单是要处理文件的,这是最基本的东西，很多人会忘记然而当上传出错后则去找程序的错误，却忘了这一点

三、编写上传控制类  
  
1、创建一个控制类: FileUploadController和一个返回结果的页面list.jsp  
2、编写提交表单的action:

**[java]** [view plain](http://blog.csdn.net/swingpyzf/article/details/20230865) [copy](http://blog.csdn.net/swingpyzf/article/details/20230865)

[print](http://blog.csdn.net/swingpyzf/article/details/20230865)[?](http://blog.csdn.net/swingpyzf/article/details/20230865)[在CODE上查看代码片](https://code.csdn.net/snippets/212406)

1. //通过Spring的autowired注解获取spring默认配置的request
2. @Autowired
3. **private** HttpServletRequest request;
5. /\*\*\*
6. \* 上传文件 用@RequestParam注解来指定表单上的file为MultipartFile
7. \*
8. \* @param file
9. \* @return
10. \*/
11. @RequestMapping("fileUpload")
12. **public** String fileUpload(@RequestParam("file") MultipartFile file) {
13. // 判断文件是否为空
14. **if** (!file.isEmpty()) {
15. **try** {
16. // 文件保存路径
17. String filePath = request.getSession().getServletContext().getRealPath("/") + "upload/"
18. + file.getOriginalFilename();
19. // 转存文件
20. file.transferTo(**new** File(filePath));
21. } **catch** (Exception e) {
22. e.printStackTrace();
23. }
24. }
25. // 重定向
26. **return** "redirect:/list.html";
27. }
29. /\*\*\*
30. \* 读取上传文件中得所有文件并返回
31. \*
32. \* @return
33. \*/
34. @RequestMapping("list")
35. **public** ModelAndView list() {
36. String filePath = request.getSession().getServletContext().getRealPath("/") + "upload/";
37. ModelAndView mav = **new** ModelAndView("list");
38. File uploadDest = **new** File(filePath);
39. String[] fileNames = uploadDest.list();
40. **for** (**int** i = 0; i < fileNames.length; i++) {
41. //打印出文件名
42. System.out.println(fileNames[i]);
43. }
44. **return** mav;
45. }

3、使用SpringMVC注解RequestParam来指定表单中的file参数；  
4、指定一个用于保存文件的web项目路径  
5、通过MultipartFile的transferTo(File dest)这个方法来转存文件到指定的路径。  
  
  
到此基本的文件上传就结束了。  
  
  
MultipartFile类常用的一些方法：  
  
  
String getContentType()//获取文件MIME类型  
InputStream getInputStream()//后去文件流  
String getName() //获取表单中文件组件的名字  
String getOriginalFilename() //获取上传文件的原名  
long getSize()  //获取文件的字节大小，单位byte  
boolean isEmpty() //是否为空  
void transferTo(File dest)

//保存到一个目标文件中。  
  
  
四、多文件上传。  
  
多文件上传其实很简单，和上传其他相同的参数如checkbox一样，表单中使用相同的名称，然后action中将MultipartFile参数类定义为数组就可以。  
接下来实现：

1、创建一个上传多文件的表单：

**[html]** [view plain](http://blog.csdn.net/swingpyzf/article/details/20230865) [copy](http://blog.csdn.net/swingpyzf/article/details/20230865)

[print](http://blog.csdn.net/swingpyzf/article/details/20230865)[?](http://blog.csdn.net/swingpyzf/article/details/20230865)[在CODE上查看代码片](https://code.csdn.net/snippets/212406)

1. **<body>**
2. **<h2>**上传多个文件 实例**</h2>**

5. **<form** action="filesUpload.html" method="post"
6. enctype="multipart/form-data"**>**
7. **<p>**
8. 选择文件:**<input** type="file" name="files"**>**
9. **<p>**
10. 选择文件:**<input** type="file" name="files"**>**
11. **<p>**
12. 选择文件:**<input** type="file" name="files"**>**
13. **<p>**
14. **<input** type="submit" value="提交"**>**
15. **</form>**
16. **</body>**

2、编写处理表单的action，将原来保存文件的方法单独写一个方法出来方便共用：

**[java]** [view plain](http://blog.csdn.net/swingpyzf/article/details/20230865) [copy](http://blog.csdn.net/swingpyzf/article/details/20230865)

[print](http://blog.csdn.net/swingpyzf/article/details/20230865)[?](http://blog.csdn.net/swingpyzf/article/details/20230865)[在CODE上查看代码片](https://code.csdn.net/snippets/212406)

1. /\*\*\*
2. \* 保存文件
3. \* @param file
4. \* @return
5. \*/
6. **private** **boolean** saveFile(MultipartFile file) {
7. // 判断文件是否为空
8. **if** (!file.isEmpty()) {
9. **try** {
10. // 文件保存路径
11. String filePath = request.getSession().getServletContext().getRealPath("/") + "upload/"
12. + file.getOriginalFilename();
13. // 转存文件
14. file.transferTo(**new** File(filePath));
15. **return** **true**;
16. } **catch** (Exception e) {
17. e.printStackTrace();
18. }
19. }
20. **return** **false**;
21. }
22. 3、编写action:
23. @RequestMapping("filesUpload")
24. **public** String filesUpload(@RequestParam("files") MultipartFile[] files) {
25. //判断file数组不能为空并且长度大于0
26. **if**(files!=**null**&&files.length>0){
27. //循环获取file数组中得文件
28. **for**(**int** i = 0;i<files.length;i++){
29. MultipartFile file = files[i];
30. //保存文件
31. saveFile(file);
32. }
33. }
34. // 重定向
35. **return** "redirect:/list.html";
36. }

最后运行项目上传文件:

