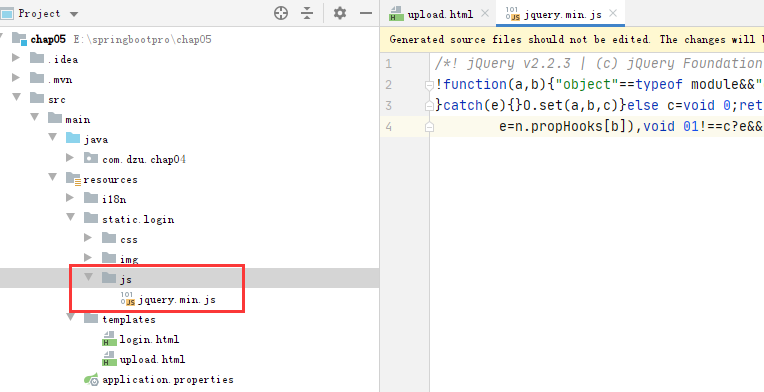
# 实验十二 文件上传与下载

一、文件上传

1. 在login文件夹中导入jquery.min.js文件

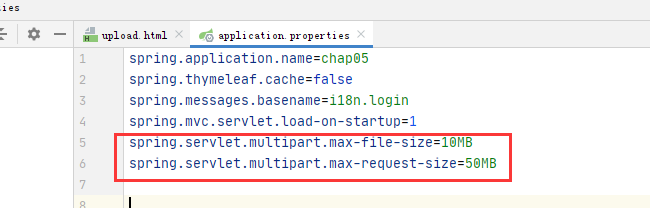


2.编写文件上传的表单页面

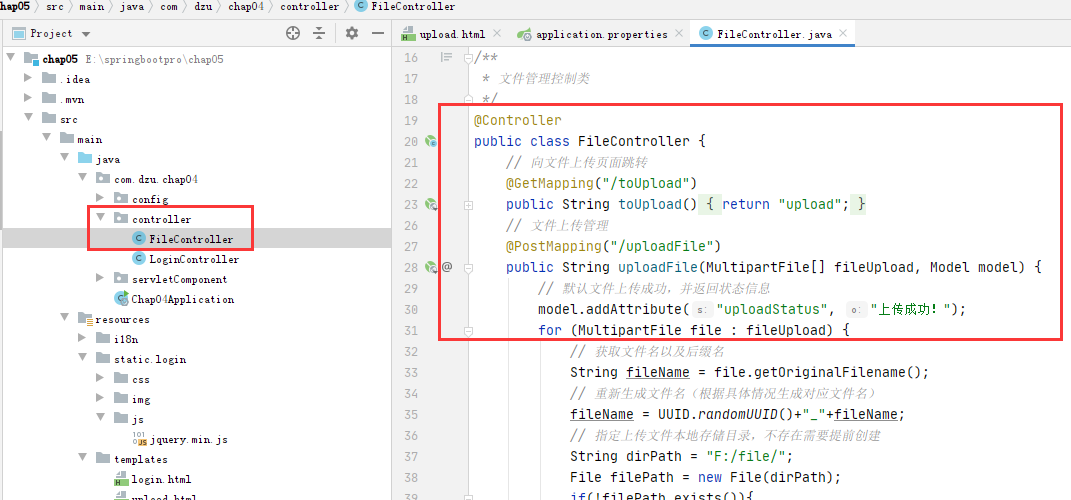


<!DOCTYPE html>  
<html lang="en" xmlns:th="http://www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
 <title>动态添加文件上传列表</title>  
 <link th:href="@{/login/css/bootstrap.min.css}" rel="stylesheet">  
 <script th:src="@{/login/js/jquery.min.js}"></script>  
</head>  
<body>  
<div th:if="${uploadStatus}" style="color: red" th:text="${uploadStatus}">上传成功</div>  
<form th:action="@{/uploadFile}" method="post" enctype="multipart/form-data">  
 上传文件:&nbsp;&nbsp;<input type="button" value="添加文件" onclick="*add*()"/>  
 <div id="file" style="margin-top: 10px;" th:value="文件上传区域"> </div>  
 <input id="submit" type="submit" value="上传"  
 style="display: none;margin-top: 10px;"/>  
</form>  
<script type="text/javascript">  
 *// 动态添加上传按钮* function *add*(){  
 var innerdiv = "<div>";  
 innerdiv += "<input type='file' name='fileUpload' required='required'>" +  
 "<input type='button' value='删除' onclick='*remove*(this)'>";  
 innerdiv +="</div>";  
 $("#file").append(innerdiv);  
 *// 打开上传按钮* $("#submit").css("display","block");  
 }  
 *// 删除当前行<div>* function *remove*(obj) {  
 $(obj).parent().remove();  
 if($("#file div").length ==0){  
 $("#submit").css("display","none");  
 }  
 }  
</script>  
</body>  
</html>

3.修改全局配置文件



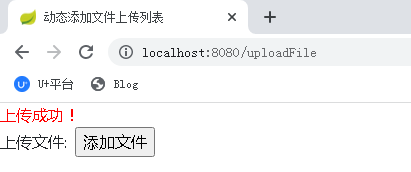
4.创建文件管理控制类

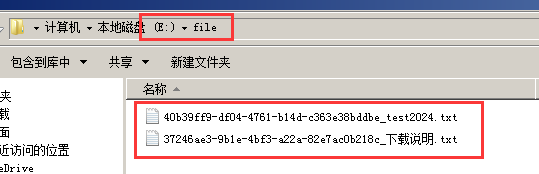


@Controller  
public class FileController {  
 *// 向文件上传页面跳转* @GetMapping("/toUpload")  
 public String toUpload(){  
 return "upload";  
 }  
 *// 文件上传管理* @PostMapping("/uploadFile")  
 public String uploadFile(MultipartFile[] fileUpload, Model model) {  
 *// 默认文件上传成功，并返回状态信息* model.addAttribute("uploadStatus", "上传成功！");  
 for (MultipartFile file : fileUpload) {  
 *// 获取文件名以及后缀名* String fileName = file.getOriginalFilename();  
 *// 重新生成文件名（根据具体情况生成对应文件名）* fileName = UUID.*randomUUID*()+"\_"+fileName;  
 *// 指定上传文件本地存储目录，不存在需要提前创建* String dirPath = "E:/file/";;  
 File filePath = new File(dirPath);  
 if(!filePath.exists()){  
 filePath.mkdirs();  
 }  
 try {  
 file.transferTo(new File(dirPath+fileName));  
 } catch (Exception e) {  
 e.printStackTrace();  
 *// 上传失败，返回失败信息* model.addAttribute("uploadStatus","上传失败： "+e.getMessage());  
 }  
 }  
 *// 携带上传状态信息回调到文件上传页面* return "upload";  
 }  
}

5.测试

运行项目启动类Chap04Application

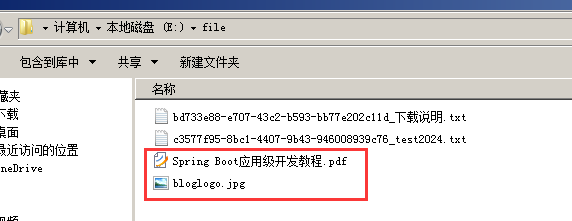


二、英文名文件下载

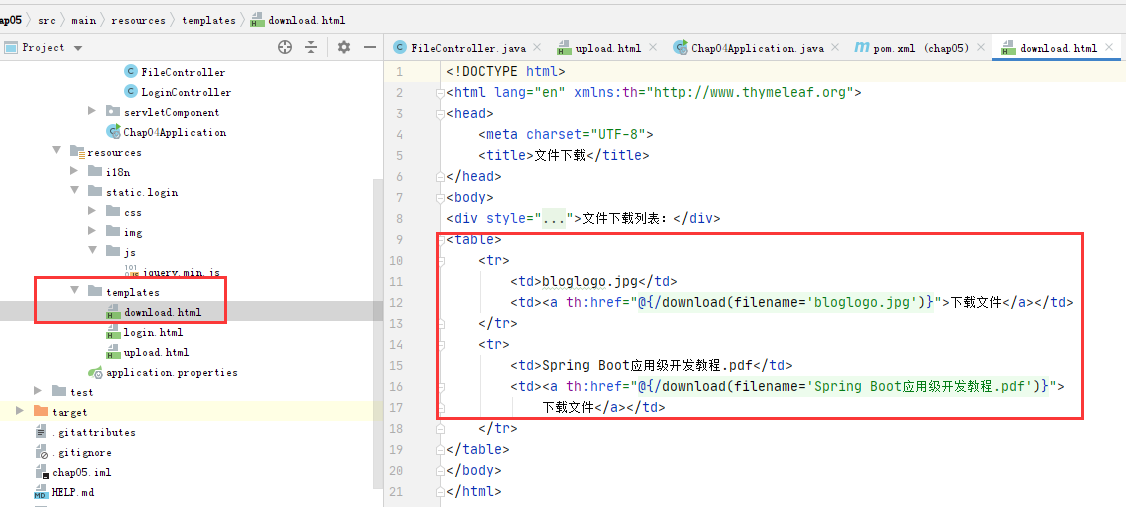
1.添加工具依赖



2.在E:\file文件夹中放入文件Spring Boot应用级开发教程.pdf和bloglogo.jpg

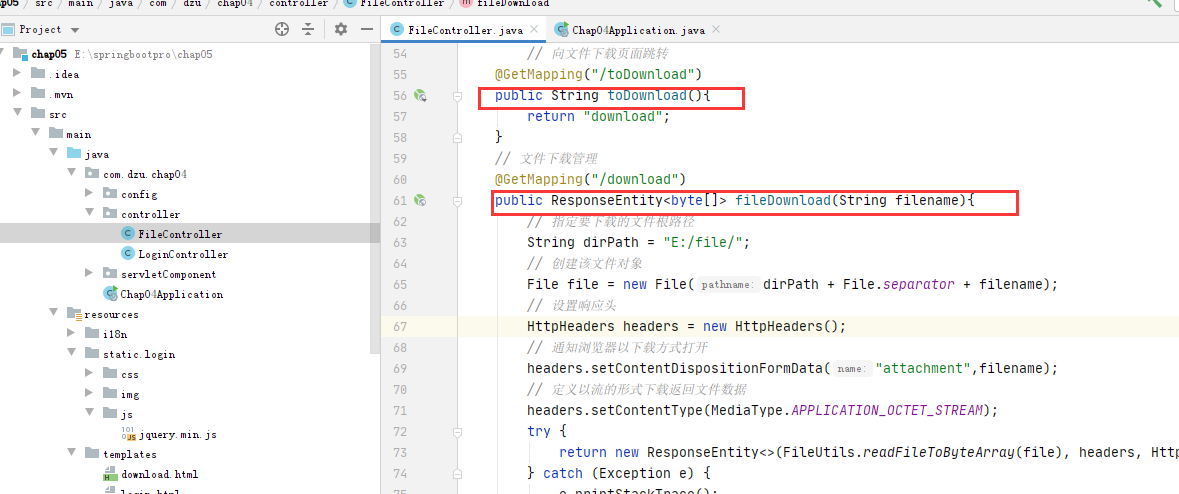


3．创建下载页面download.html



<!DOCTYPE html>  
<html lang="en" xmlns:th="http://www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title>文件下载</title>  
</head>  
<body>  
<div style="margin-bottom: 10px">文件下载列表：</div>  
<table>  
 <tr>  
 <td>bloglogo.jpg</td>  
 <td><a th:href="@{/download(filename='bloglogo.jpg')}">下载文件</a></td>  
 </tr>  
 <tr>  
 <td>Spring Boot应用级开发教程.pdf</td>  
 <td><a th:href="@{/download(filename='Spring Boot应用级开发教程.pdf')}">  
 下载文件</a></td>  
 </tr>  
</table>  
</body>  
</html>

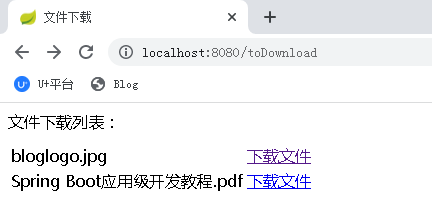
4．在FileController类中添加文件下载处理方法

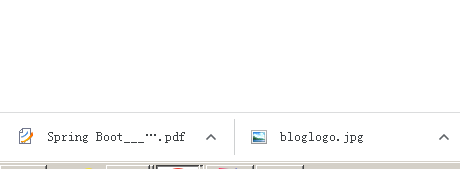


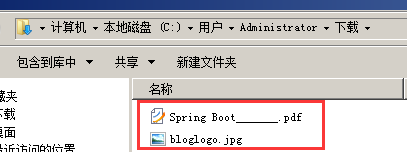
*// 向文件下载页面跳转*@GetMapping("/toDownload")  
public String toDownload(){  
 return "download";  
}  
*// 文件下载管理*@GetMapping("/download")  
public ResponseEntity<byte[]> fileDownload(String filename){  
 *// 指定要下载的文件根路径* String dirPath = "E:/file/";  
 *// 创建该文件对象* File file = new File(dirPath + File.*separator* + filename);  
 *// 设置响应头* HttpHeaders headers = new HttpHeaders();  
 *// 通知浏览器以下载方式打开* headers.setContentDispositionFormData("attachment",filename);  
 *// 定义以流的形式下载返回文件数据* headers.setContentType(MediaType.*APPLICATION\_OCTET\_STREAM*);  
 try {  
 return new ResponseEntity<>(FileUtils.*readFileToByteArray*(file), headers, HttpStatus.*OK*);  
 } catch (Exception e) {  
 e.printStackTrace();  
 return new ResponseEntity<byte[]>(e.getMessage().getBytes(),HttpStatus.*EXPECTATION\_FAILED*);  
 }  
}

4.测试

重启项目启动类Chap04Application

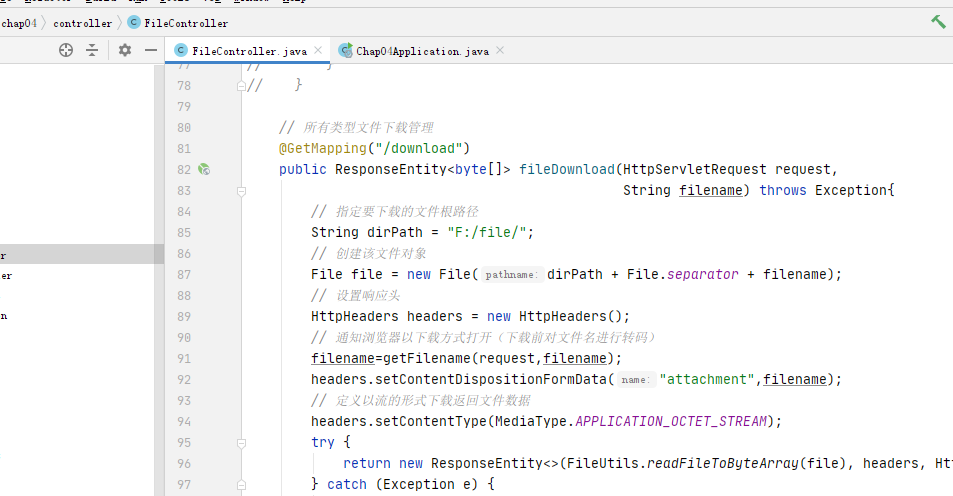






三、中文名文件下载

1．修改FileContrller类的fileDownload()方法



*// 所有类型文件下载管理*@GetMapping("/download")  
public ResponseEntity<byte[]> fileDownload(HttpServletRequest request,  
 String filename) throws Exception{  
 *// 指定要下载的文件根路径* String dirPath = "E:/file/";  
 *// 创建该文件对象* File file = new File(dirPath + File.*separator* + filename);  
 *// 设置响应头* HttpHeaders headers = new HttpHeaders();  
 *// 通知浏览器以下载方式打开（下载前对文件名进行转码）* filename=getFilename(request,filename);  
 headers.setContentDispositionFormData("attachment",filename);  
 *// 定义以流的形式下载返回文件数据* headers.setContentType(MediaType.*APPLICATION\_OCTET\_STREAM*);  
 try {  
 return new ResponseEntity<>(FileUtils.*readFileToByteArray*(file), headers, HttpStatus.*OK*);  
 } catch (Exception e) {  
 e.printStackTrace();  
 return new ResponseEntity<byte[]>(e.getMessage().getBytes(),HttpStatus.*EXPECTATION\_FAILED*);  
 }  
}  
*// 根据浏览器的不同进行编码设置，返回编码后的文件名*private String getFilename(HttpServletRequest request, String filename)  
 throws Exception {  
 *// IE不同版本User-Agent中出现的关键词* String[] IEBrowserKeyWords = {"MSIE", "Trident", "Edge"};  
 *// 获取请求头代理信息* String userAgent = request.getHeader("User-Agent");  
 for (String keyWord : IEBrowserKeyWords) {  
 if (userAgent.contains(keyWord)) {  
 *//IE内核浏览器，统一为UTF-8编码显示，并对转换的+进行更正* return URLEncoder.*encode*(filename, "UTF-8").replace("+"," ");  
 }  
 }  
 *//火狐等其它浏览器统一为ISO-8859-1编码显示* return new String(filename.getBytes("UTF-8"), "ISO-8859-1");  
}

2．测试

重启项目启动类

