上传文件

1. 引入依赖

fastdfs

1. 引入工具类FastDFSClient
2. 在shop工程中依赖common并引入fastdfs配置文件,引入common-fileupload
3. Springmvc.xml文件中配置文件上传解析器
4. 编写uploadController

public class ImageUploadController {  
  
 @Value("${SERVER\_ID}")  
 private String fastdfsIp;  
 @RequestMapping("/imageUpload")  
 public Result imageUpload(MultipartFile uploadFile) throws Exception {  
  
 try {  
String extName=uploadFile.getOriginalFilename().substring(uploadFile.getOriginalFilename().lastIndexOf(".")+1);  
 FastDFSClient fastDFSClient=new FastDFSClient("classpath:config/fdfs\_client.conf");  
 String url = fastDFSClient.uploadFile(uploadFile.getBytes(), extName);  
   
 return new Result(true,fastdfsIp+url);  
 } catch (Exception e) {  
  
 e.printStackTrace();  
 return new Result(false,"失败");  
 }  
  
 }  
}

注入fastdfs服务器的地址

截取扩展名

调用工具类将文件保存到fastdfs服务器并获得保存路径

将服务器地址与保存路径拼接返回给前端

前端编写

1. service层提交方式

调用formData()

formData.append(“key”,doucument.getElementById(“”).files[0]) //获取文件上传表单项的第1个文件 如果加了 multi 表示上传多个文件

$http({

url:

method:post

data:formatData

headers:{“Content-Type”:undified}

transforRequest:angular.identify

})