文件上传

配置文件dev

**spring:** *#-------------file----------#* **http:  
 multipart:  
 enabled:** true *#支持文件上传  
 #location: E:/mycode/springboot/tempfile #这个目录程序不会自己创建，可以不配置，*

**file-size-threshold:** 0 *#文件大小阈值，当大于这个阈值时将写入到磁盘，否则在内存中。默认值为0* **max-file-size:** 1024KB *#文件最大 1MB* **max-request-size:** 1536KB *#请求总体大小,1.5MB*

controller

import com.first.study.service.service.fileupload.service.FileUploadService;  
import org.springframework.beans.factory.annotation.**Autowired**;  
import org.springframework.stereotype.**Controller**;  
import org.springframework.web.bind.annotation.**PostMapping**;  
import org.springframework.web.bind.annotation.**RequestMapping**;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.**ResponseBody**;  
import org.springframework.web.multipart.MultipartFile;**@Controller  
@RequestMapping**(value = **"/api/file/"**)  
public class TestFileUploadController {  
 **@Autowired** FileUploadService fileUploadService;  
 //@RequestMapping(value = "uploadFile",method = RequestMethod.POST)  
 **@PostMapping**(value = **"uploadFile"**)  
 **@ResponseBody** public String uploadFile(MultipartFile file){  
 fileUploadService.store(file);  
 return **"over"**;  
 }  
}

service

import org.springframework.core.io.Resource;  
import org.springframework.web.multipart.MultipartFile;public interface FileUploadService {  
 void store(MultipartFile file);  
}

serviceImpl

import org.springframework.core.io.Resource;  
import org.springframework.stereotype.**Service**;  
import org.springframework.util.FileSystemUtils;  
import org.springframework.util.StringUtils;  
import org.springframework.web.multipart.MultipartFile;  
import java.io.IOException;  
import java.io.InputStream;  
import java.nio.file.Files;  
import java.nio.file.Path;  
import java.nio.file.Paths;  
import java.nio.file.StandardCopyOption;**@Service**public class FileUploadServiceImpl implements FileUploadService {  
 Path rootLocation= Paths.*get*(**"E:/mycode/"**);  
 **@Override** public void store(MultipartFile file) {  
 String filename = StringUtils.*cleanPath*(file.getOriginalFilename());  
 try (InputStream inputStream = file.getInputStream()) {  
 Files.*copy*(inputStream, this.rootLocation.resolve(filename), StandardCopyOption.*REPLACE\_EXISTING*);  
 } catch (IOException e) {  
 System.*err*.println(**"store----------------"**);  
 }  
 }

我发现配置文件location，运行时controller的断点到service，此时可以看到location目录下生成一个临时文件，（不指定location肯定也会有临时文件，只是我没找到位置）

这个临时文件夹要不要指定，有什么影响。

**客户端文件---服务器临时文件夹-----指定文件上传目录**