



链滴

使用 MultipartFile 报错：java.io.IOException: java.io.FileNotFoundException(系统找不到指定的路径。)

作者：[moyv](#)

原文链接：<https://ld246.com/article/1587547355960>

来源网站：[链滴](#)

许可协议：[署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

使用MultipartFile报错: java.io.IOException: java.io.FileNotFoundException(系统找不到指定的路径。)

在使用公司框架开发时遇到一个问题: spring框架MultipartFile上传文件报错。

开头说问题原因: MultipartFile的transferTo()方法会检查传入其中的File对象的路径是否为绝对路, 如果不是绝对路径, 会自动拼接application里的预设路径和\work\Tomcat\localhost\ROOT

解决过程:

先看代码:

```
public Map upload( HttpServletRequest request, HttpServletResponse response, @RequestParam("file")MultipartFile upload) throws IllegalStateException, IOException {
    String realPath = request.getParameter("target");//userfiles\1\程序附件\theFile\problem\
    if (!upload.isEmpty()) {
        //获取原文件名
        String oldName=upload.getOriginalFilename();
        //获取文件大小: 字节
        long size=upload.getSize();
        //获取文件后缀名
        String format=oldName.substring(oldName.lastIndexOf(".")+1);
        //使用随机新文件名,
        String newName=IdGen.uuid()+"\\"+format;
        //创建目录
        FileUtils.createDirectory(realPath);
        File file = FileUtils.getAvailableFile(realPath+newName, 0);
        upload.transferTo(file);//这里报错
    }
    return null;
}
```

报错信息:

```
java.io.IOException: java.io.FileNotFoundException: E:\Administrator\Documents\Java\Project\boot\work\Tomcat\localhost\ROOT\userfiles\1\程序附件\theFile\problem\882b77ffb7f74e694cc1cf909fd3b59.mp3 (系统找不到指定的路径。)
    at org.apache.catalina.core.ApplicationPart.write(ApplicationPart.java:122) ~[tomcat-embed-core-8.5.32.jar:8.5.32]
    at org.springframework.web.multipart.support.StandardMultipartHttpServletRequest$StandardMultipartFile.transferTo(StandardMultipartHttpServletRequest.java:255) ~[spring-web-5.0.8.RELEASE.jar:5.0.8.RELEASE]
    at com.jeeplus.modules.file.web.TheFileController.upload(TheFileController.java:284) ~[classes/:na]
    at com.jeeplus.modules.file.web.TheFileController$$FastClassBySpringCGLIB$$b6b3e23b.invoke(<generated>) ~[classes/:na]
    at org.springframework.cglib.proxy.MethodProxy.invoke(MethodProxy.java:204) ~[spring-core-5.0.8.RELEASE.jar:5.0.8.RELEASE]
    at org.springframework.aop.framework.CglibAopProxy$DynamicAdvisedInterceptor.intercept(CglibAopProxy.java:684) ~[spring-aop-5.0.8.RELEASE.jar:5.0.8.RELEASE]
    at com.jeeplus.modules.file.web.TheFileController$$EnhancerBySpringCGLIB$$ce883134.upload(<generated>) ~[classes/:na]
    at com.jeeplus.modules.file.web.TheFileController$$FastClassBySpringCGLIB$$b6b3e23b.in
```

```
oke(<generated>) ~[classes/:na]
  at org.springframework.cglib.proxy.MethodProxy.invoke(MethodProxy.java:204) ~[spring-core-5.0.8.RELEASE.jar:5.0.8.RELEASE]
  at org.springframework.aop.framework.CglibAopProxy$DynamicAdvisedInterceptor.intercept(CglibAopProxy.java:684) ~[spring-aop-5.0.8.RELEASE.jar:5.0.8.RELEASE]
  at com.jeeplus.modules.file.web.TheFileController$$EnhancerBySpringCGLIB$$e3edb904.upload(<generated>) ~[classes/:na]
  at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) ~[na:1.8.0_131]
  at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source) ~[na:1.8.0_131]
  at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source) ~[na:1.8.0_131]
  at java.lang.reflect.Method.invoke(Unknown Source) ~[na:1.8.0_131]
  at org.springframework.web.method.support.InvocableHandlerMethod.doInvoke(InvocableHandlerMethod.java:209) ~[spring-web-5.0.8.RELEASE.jar:5.0.8.RELEASE]
  at org.springframework.web.method.support.InvocableHandlerMethod.invokeForRequest(InvocableHandlerMethod.java:136) ~[spring-web-5.0.8.RELEASE.jar:5.0.8.RELEASE]
  at org.springframework.web.servlet.mvc.method.annotation.ServletInvocableHandlerMethod.invokeAndHandle(ServletInvocableHandlerMethod.java:102) ~[spring-webmvc-5.0.8.RELEASE.jar:5.0.8.RELEASE]
  at org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.invokeHandlerMethod(RequestMappingHandlerAdapter.java:877) ~[spring-webmvc-5.0.8.RELEASE.jar:5.0.8.RELEASE]
  at org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.handleInternal(RequestMappingHandlerAdapter.java:783) ~[spring-webmvc-5.0.8.RELEASE.jar:5.0.8.RELEASE]
  at org.springframework.web.servlet.mvc.method.AbstractHandlerMethodAdapter.handle(AbstractHandlerMethodAdapter.java:87) ~[spring-webmvc-5.0.8.RELEASE.jar:5.0.8.RELEASE]
  at org.springframework.web.servlet.DispatcherServlet.doDispatch(DispatcherServlet.java:99) ~[spring-webmvc-5.0.8.RELEASE.jar:5.0.8.RELEASE]
  at org.springframework.web.servlet.DispatcherServlet.doService(DispatcherServlet.java:925) ~[spring-webmvc-5.0.8.RELEASE.jar:5.0.8.RELEASE]
  at org.springframework.web.servlet.FrameworkServlet.processRequest(FrameworkServlet.java:974) [spring-webmvc-5.0.8.RELEASE.jar:5.0.8.RELEASE]
  at org.springframework.web.servlet.FrameworkServlet.doPost(FrameworkServlet.java:877) [spring-webmvc-5.0.8.RELEASE.jar:5.0.8.RELEASE]
  at javax.servlet.http.HttpServlet.service(HttpServlet.java:661) [tomcat-embed-core-8.5.32.jar:8.5.32]
  at org.springframework.web.servlet.FrameworkServlet.service(FrameworkServlet.java:851) [spring-webmvc-5.0.8.RELEASE.jar:5.0.8.RELEASE]
  at javax.servlet.http.HttpServlet.service(HttpServlet.java:742) [tomcat-embed-core-8.5.32.jar:8.5.32]
  at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:231) [tomcat-embed-core-8.5.32.jar:8.5.32]
  at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166) [tomcat-embed-core-8.5.32.jar:8.5.32]
  at org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:52) [tomcat-embed-socket-8.5.32.jar:8.5.32]
  at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:193) [tomcat-embed-core-8.5.32.jar:8.5.32]
  at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166) [tomcat-embed-core-8.5.32.jar:8.5.32]
  at org.apache.shiro.web.servlet.AdviceFilter.executeChain(AdviceFilter.java:108) [shiro-web-1.4.0-RC2.jar:1.4.0-RC2]
  at org.apache.shiro.web.servlet.AdviceFilter.doFilterInternal(AdviceFilter.java:137) [shiro-web-1.4.0-RC2.jar:1.4.0-RC2]
```

```
at org.apache.shiro.web.servlet.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:12
) [shiro-web-1.4.0-RC2.jar:1.4.0-RC2]
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.ja
a:193) [tomcat-embed-core-8.5.32.jar:8.5.32]
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166) [
omcat-embed-core-8.5.32.jar:8.5.32]
at org.apache.shiro.web.servlet.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:11
) [shiro-web-1.4.0-RC2.jar:1.4.0-RC2]
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.ja
a:193) [tomcat-embed-core-8.5.32.jar:8.5.32]
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166) [
omcat-embed-core-8.5.32.jar:8.5.32]
at org.apache.shiro.web.servlet.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:11
) [shiro-web-1.4.0-RC2.jar:1.4.0-RC2]
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.ja
a:193) [tomcat-embed-core-8.5.32.jar:8.5.32]
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166) [
omcat-embed-core-8.5.32.jar:8.5.32]
at org.apache.shiro.web.servlet.AdviceFilter.executeChain(AdviceFilter.java:108) [shiro-web-
.4.0-RC2.jar:1.4.0-RC2]
at org.apache.shiro.web.servlet.AdviceFilter.doFilterInternal(AdviceFilter.java:137) [shiro-we
-1.4.0-RC2.jar:1.4.0-RC2]
at org.apache.shiro.web.servlet.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:12
) [shiro-web-1.4.0-RC2.jar:1.4.0-RC2]
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.ja
a:193) [tomcat-embed-core-8.5.32.jar:8.5.32]
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166) [
omcat-embed-core-8.5.32.jar:8.5.32]
at com.alibaba.druid.support.http.WebStatFilter.doFilter(WebStatFilter.java:123) [druid-1.1.
0.jar:1.1.10]
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.ja
a:193) [tomcat-embed-core-8.5.32.jar:8.5.32]
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166) [
omcat-embed-core-8.5.32.jar:8.5.32]
at org.apache.shiro.web.servlet.ProxiedFilterChain.doFilter(ProxiedFilterChain.java:61) [shiro
web-1.4.0-RC2.jar:1.4.0-RC2]
at org.apache.shiro.web.servlet.AdviceFilter.executeChain(AdviceFilter.java:108) [shiro-web-
.4.0-RC2.jar:1.4.0-RC2]
at org.apache.shiro.web.servlet.AdviceFilter.doFilterInternal(AdviceFilter.java:137) [shiro-we
-1.4.0-RC2.jar:1.4.0-RC2]
at org.apache.shiro.web.servlet.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:12
) [shiro-web-1.4.0-RC2.jar:1.4.0-RC2]
at org.apache.shiro.web.servlet.ProxiedFilterChain.doFilter(ProxiedFilterChain.java:66) [shiro
web-1.4.0-RC2.jar:1.4.0-RC2]
at org.apache.shiro.web.servlet.AdviceFilter.executeChain(AdviceFilter.java:108) [shiro-web-
.4.0-RC2.jar:1.4.0-RC2]
at org.apache.shiro.web.servlet.AdviceFilter.doFilterInternal(AdviceFilter.java:137) [shiro-we
-1.4.0-RC2.jar:1.4.0-RC2]
at org.apache.shiro.web.servlet.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:12
) [shiro-web-1.4.0-RC2.jar:1.4.0-RC2]
at org.apache.shiro.web.servlet.ProxiedFilterChain.doFilter(ProxiedFilterChain.java:66) [shiro
web-1.4.0-RC2.jar:1.4.0-RC2]
at org.apache.shiro.web.servlet.AbstractShiroFilter.executeChain(AbstractShiroFilter.java:449
) [shiro-web-1.4.0-RC2.jar:1.4.0-RC2]
```

```
at org.apache.shiro.web.servlet.AbstractShiroFilter$1.call(AbstractShiroFilter.java:365) [shiro-web-1.4.0-RC2.jar:1.4.0-RC2]
at org.apache.shiro.subject.support.SubjectCallable.doCall(SubjectCallable.java:90) [shiro-core-1.4.0-RC2.jar:1.4.0-RC2]
at org.apache.shiro.subject.support.SubjectCallable.call(SubjectCallable.java:83) [shiro-core-1.4.0-RC2.jar:1.4.0-RC2]
at org.apache.shiro.subject.support.DelegatingSubject.execute(DelegatingSubject.java:387) [shiro-core-1.4.0-RC2.jar:1.4.0-RC2]
at org.apache.shiro.web.servlet.AbstractShiroFilter.doFilterInternal(AbstractShiroFilter.java:32) [shiro-web-1.4.0-RC2.jar:1.4.0-RC2]
at org.apache.shiro.web.servlet.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:125) [shiro-web-1.4.0-RC2.jar:1.4.0-RC2]
at org.springframework.web.filter.DelegatingFilterProxy.invokeDelegate(DelegatingFilterProxy.java:357) [spring-web-5.0.8.RELEASE.jar:5.0.8.RELEASE]
at org.springframework.web.filter.DelegatingFilterProxy.doFilter(DelegatingFilterProxy.java:70) [spring-web-5.0.8.RELEASE.jar:5.0.8.RELEASE]
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:193) [tomcat-embed-core-8.5.32.jar:8.5.32]
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166) [tomcat-embed-core-8.5.32.jar:8.5.32]
at org.springframework.web.filter.CharacterEncodingFilter.doFilterInternal(CharacterEncodingFilter.java:200) [spring-web-5.0.8.RELEASE.jar:5.0.8.RELEASE]
at org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:97) [spring-web-5.0.8.RELEASE.jar:5.0.8.RELEASE]
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:193) [tomcat-embed-core-8.5.32.jar:8.5.32]
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166) [tomcat-embed-core-8.5.32.jar:8.5.32]
at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:198) [tomcat-embed-core-8.5.32.jar:8.5.32]
at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:96) [tomcat-embed-core-8.5.32.jar:8.5.32]
at org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:493) [tomcat-embed-core-8.5.32.jar:8.5.32]
at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:140) [tomcat-embed-core-8.5.32.jar:8.5.32]
at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:81) [tomcat-embed-core-8.5.32.jar:8.5.32]
at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:87) [tomcat-embed-core-8.5.32.jar:8.5.32]
at org.apache.catalina.valves.AbstractAccessLogValve.invoke(AbstractAccessLogValve.java:50) [tomcat-embed-core-8.5.32.jar:8.5.32]
at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:342) [tomcat-embed-core-8.5.32.jar:8.5.32]
at org.apache.coyote.http11.Http11Processor.service(Http11Processor.java:800) [tomcat-embed-core-8.5.32.jar:8.5.32]
at org.apache.coyote.AbstractProcessorLight.process(AbstractProcessorLight.java:66) [tomcat-embed-core-8.5.32.jar:8.5.32]
at org.apache.coyote.AbstractProtocol$ConnectionHandler.process(AbstractProtocol.java:80) [tomcat-embed-core-8.5.32.jar:8.5.32]
at org.apache.tomcat.util.net.NioEndpoint$SocketProcessor.doRun(NioEndpoint.java:1471) [tomcat-embed-core-8.5.32.jar:8.5.32]
at org.apache.tomcat.util.net.SocketProcessorBase.run(SocketProcessorBase.java:49) [tomcat-embed-core-8.5.32.jar:8.5.32]
```

```
at java.util.concurrent.ThreadPoolExecutor.runWorker(Unknown Source) [na:1.8.0_131]
at java.util.concurrent.ThreadPoolExecutor$Worker.run(Unknown Source) [na:1.8.0_131]
at org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61) [
omcat-embed-core-8.5.32.jar:8.5.32]
at java.lang.Thread.run(Unknown Source) [na:1.8.0_131]
Caused by: java.io.FileNotFoundException: E:\Administrator\Documents\Java\Project4\boot\
ork\Tomcat\localhost\ROOT\server.tomcat.basedir\userfiles\1\程序附件\theFile\problem\882
77ffb7f74e6984cc1cf909fd3b59.mp3 (系统找不到指定的路径。)
at java.io.FileOutputStream.open0(Native Method) ~[na:1.8.0_131]
at java.io.FileOutputStream.open(Unknown Source) ~[na:1.8.0_131]
at java.io.FileOutputStream.<init>(Unknown Source) ~[na:1.8.0_131]
at java.io.FileOutputStream.<init>(Unknown Source) ~[na:1.8.0_131]
at org.apache.tomcat.util.http.fileupload.disk.DiskFileItem.write(DiskFileItem.java:400) ~[to
cat-embed-core-8.5.32.jar:8.5.32]
at org.apache.catalina.core.ApplicationPart.write(ApplicationPart.java:120) ~[tomcat-embed
core-8.5.32.jar:8.5.32]
... 92 common frames omitted
```

分析:

首先, transferTo()方法来自jar包不太可能出错, 参数fileFileUtils来自公司框架, 先检查FileUtils代
:

```
public static File getAvailableFile(String name, int index){
    File newFile = null;
    String suffix = StringUtils.substringAfterLast(name, ".");
    String filePath = StringUtils.substringBeforeLast(name, ".");
    if(index == 0){
        newFile = new File(filePath+"."+suffix);
    }else{
        newFile = new File(filePath+"("+index+")"+"."+suffix);
    }
    if(newFile.exists()){
        return getAvailableFile(name,index+1);
    }else{
        return newFile;
    }
};
```

没有发现问题,

输出file的path, 没有错误

然后猜测是transferTo()方法使用方式错误, 查看报错信息, 找到抛出错误的方法

@Override

```
public void transferTo(File dest) throws IOException, IllegalStateException {
    this.part.write(dest.getPath());
    if (dest.isAbsolute() && !dest.exists()) {
        // Servlet 3.0 Part.write is not guaranteed to support absolute file paths:
        // may translate the given path to a relative location within a temp dir
```



```

        // (e.g. on Jetty whereas Tomcat and Undertow detect absolute paths).
        // At least we offloaded the file from memory storage; it'll get deleted
        // from the temp dir eventually in any case. And for our user's purposes,
        // we can manually copy it to the requested location as a fallback.
        FileCopyUtils.copy(this.part.getInputStream(), Files.newOutputStream(dest.toPath()))
    }
}

```

点击write继续深入

```

public void write(String fileName) throws IOException {
    File file = new File(fileName);
    if (!file.isAbsolute()) {
        file = new File(location, fileName);
    }
    try {
        fileItem.write(file);
    } catch (Exception e) {
        throw new IOException(e);
    }
}

```

找到原因，如果传输过来的路径是相对路径，方法会将路径拼接为绝对路径，绝对路径前半部分来自application配置文件，后半部分为tomcat的路径，所以报错。

解决方案：在将路径字符串传入transferTo()方法前，将路径拼接为绝对路径

代码：

```

@Value("${server.tomcat.basedir}")
private String systemPath;//通过注解获取application中的属性

public Map upload( HttpServletRequest request, HttpServletResponse response, @RequestParam("file")MultipartFile upload) throws IllegalStateException, IOException {
    String realPath =(systemPath.endsWith(File.separator)?systemPath:(systemPath + File.separator))+request.getParameter("target");
    if (!upload.isEmpty()) {
        //获取原文件名
        String oldName=upload.getOriginalFilename();
        //获取文件大小：字节
        long size=upload.getSize();
        //获取文件后缀名
        String format=oldName.substring(oldName.lastIndexOf(".")+1);
        //使用随机新文件名，
        String newName=IdGen.uuid()+"\\".+format;
        //创建目录
        FileUtils.createDirectory(realPath);
        File file = FileUtils.getAvailableFile(realPath+newName, 0);
        upload.transferTo(file);//这里报错
    }
    return null;
}

```