



链滴

spring-boot-devtools 引发的问题

作者: [tlbcc](#)

原文链接: <https://ld246.com/article/1633936149057>

来源网站: [链滴](#)

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

问题描述

记录本人使用spring-boot-dev-tools在开发环境进行热加载的时候，遇到的问题。

- 问题1

在使用Hutool-All依赖时调用ObjectUtil.clone()进行对象深拷贝时发生java.lang.ClassCastException问题

```
java.lang.ClassCastException: com.example.pojo.User cannot be cast to com.example.pojo.User
```

- 问题2

在使用Mybatis框架时，多模块项目中，非本项目的依赖中，使用了本项目的Mapper接口，在启动时，SpringBoot无法注入依赖，从而导致项目启动失败

```
The bean 'userMapper' could not be injected as a 'com.example.mapper.UserMapper' because it is a JDK dynamic proxy that implements:  
com.baomidou.mybatisplus.core.mapper.BaseMapper
```

解决方式

不使用spring-dev-devtools

手动添加需要热加载的三方依赖

在项目目录中添加一个配置文件 `META-INF/spring-devtools.properties`来解决这个问题

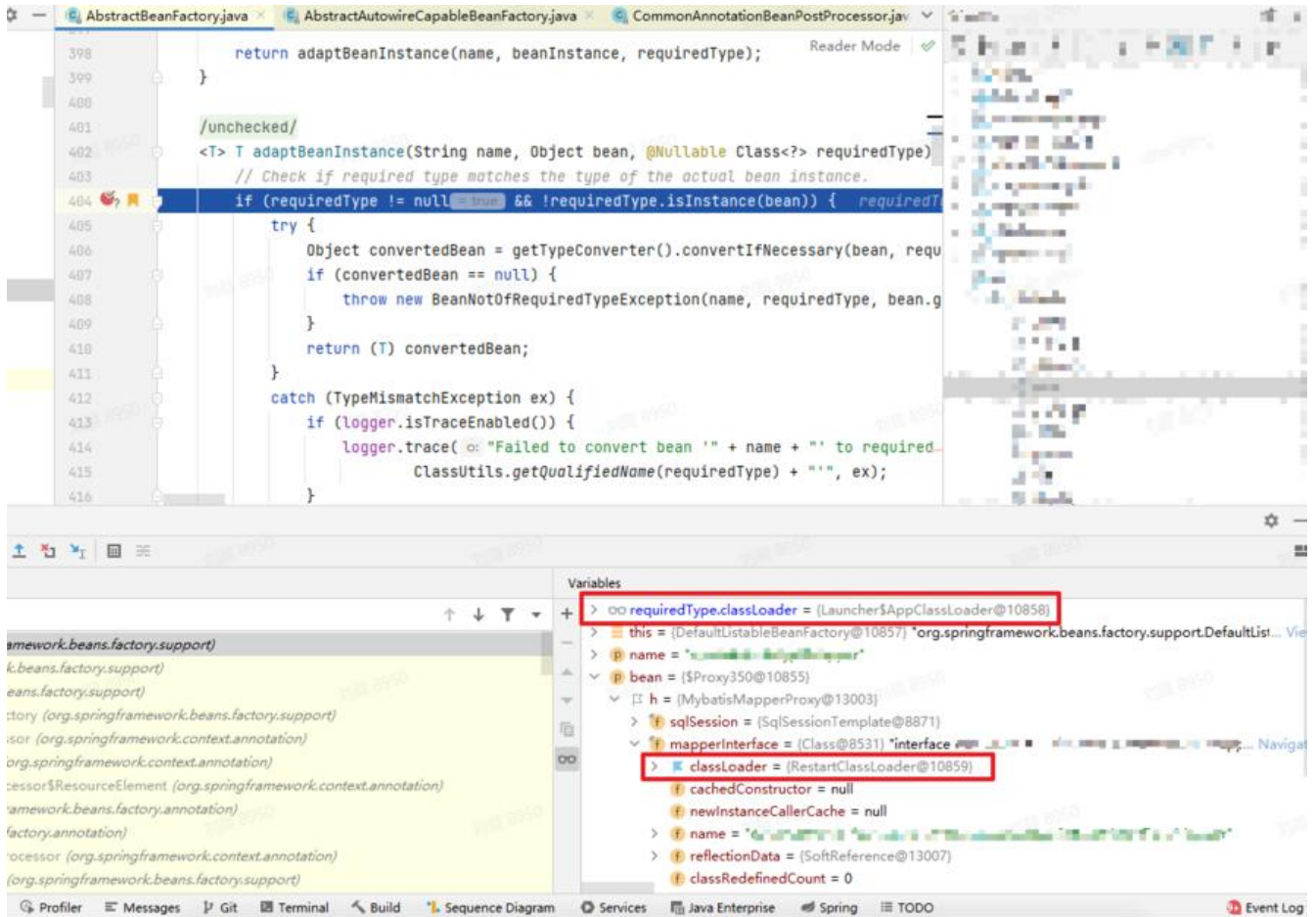
```
# 解决问题1  
# hutool-all jar包  
restart.include.hutool=/hutool-all-[\\d\\.]+\\.jar  
# 解决问题2  
# 其他模块jar包  
restart.include.other=/other-[\\d\\.]+\\.jar
```

问题产生原因

因为使用 `spring-boot-devtools`进行热加载时，仅加载本地代码的所有类，并不会加载任何依赖，且 `spring-boot-devtools`加载时会更改 `ClassLoader`，使用 `RestartClassLoader`进行类加载，导致本地代码使用了 `RestartClassLoader`加载类，第三方依赖使用了 `Launcher$AppClassLoader`加载类，同样对象在进行类型对比时（如 `instanceof` 或 `Object.class.isInstance()`）由于 `ClassLoader`不同，会判不一致，从而导致 `问题1`的对象拷贝后因类型不一致，又无法转换抛出异常，以及 `问题2`的Mybatis用JDK动态代理的Mapper接口实现类无法注入到依赖Jar包中。

问题2报错位置断点观察截图

- 由于ClassLoader不同导致判定失败



参考文章

<https://docs.spring.io/spring-boot/docs/current/reference/html/using.html#using.devtools.restart.customizing-the-classload>