



链滴

Spring4+Hibernate4 整合经验：事务配置导致的 java.lang.NoSuchMethodError 异常

作者：[CM](#)

原文链接：<https://ld246.com/article/1486375707142>

来源网站：[链滴](#)

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

最近在做项目的时候，升级之前的项目框架，从spring3+hibernate3升级到spring4+hibernate4+shiro框架的时候，遇到了一个nosuchmethod的异常问题，网上大部分资料都是说jar冲突问题，但是使用mvn控制的，冲突都解决了，仍然报这个异常，并且提示的是applicationContext.xml上下文环境配置文件里的异常异常代码如下：

```
Caused by: java.lang.NoSuchMethodError: org.springframework.aop.config.AopNamespaceUtils.registerAutoProxyCreatorIfNecessary(Lorg/springframework/beans/factory/xml/ParserContext;Ljava/lang/Object;)V
    at org.springframework.transaction.config.AnnotationDrivenBeanDefinitionParser.parseInternal(AnnotationDrivenBeanDefinitionParser.java:54) ~[spring-dao-2.0.8.jar:2.0.8]
    at org.springframework.beans.factory.xml.AbstractBeanDefinitionParser.parse(AbstractBeanDefinitionParser.java:60) ~[spring-beans-4.1.3.RELEASE.jar:4.1.3.RELEASE]
    at org.springframework.beans.factory.xml.NamespaceHandlerSupport.parse(NamespaceHandlerSupport.java:74) ~[spring-beans-4.1.3.RELEASE.jar:4.1.3.RELEASE]
    at org.springframework.beans.factory.xml.BeanDefinitionParserDelegate.parseCustomElement(BeanDefinitionParserDelegate.java:1427) ~[spring-beans-4.1.3.RELEASE.jar:4.1.3.RELEASE]
    at org.springframework.beans.factory.xml.BeanDefinitionParserDelegate.parseCustomElement(BeanDefinitionParserDelegate.java:1417) ~[spring-beans-4.1.3.RELEASE.jar:4.1.3.RELEASE]
    at org.springframework.beans.factory.xml.DefaultBeanDefinitionDocumentReader.parseBeanDefinitions(DefaultBeanDefinitionDocumentReader.java:174) ~[spring-beans-4.1.3.RELEASE.jar:4.1.3.RELEASE]
    at org.springframework.beans.factory.xml.DefaultBeanDefinitionDocumentReader.doRegisterBeanDefinitions(DefaultBeanDefinitionDocumentReader.java:144) ~[spring-beans-4.1.3.RELEASE.jar:4.1.3.RELEASE]
    at org.springframework.beans.factory.xml.DefaultBeanDefinitionDocumentReader.registerBeanDefinitions(DefaultBeanDefinitionDocumentReader.java:100) ~[spring-beans-4.1.3.RELEASE.jar:4.1.3.RELEASE]
    at org.springframework.beans.factory.xml.XmlBeanDefinitionReader.registerBeanDefinitions(XmlBeanDefinitionReader.java:510) ~[spring-beans-4.1.3.RELEASE.jar:4.1.3.RELEASE]
    at org.springframework.beans.factory.xml.XmlBeanDefinitionReader.doLoadBeanDefinitions(XmlBeanDefinitionReader.java:392) ~[spring-beans-4.1.3.RELEASE.jar:4.1.3.RELEASE]
    ... 56 common frames omitted
```

从异常上看出是registerproxycreatorifnecessary()这个方法长不到，而配置文件中，再看看自己的配置文件：

```
classpath*:jdbc.properties
```

```
classpath:sqlConfig.xml
```

```
    <property name="hibernate.driver_class" value="${jdbc.driverClassName}" />  
    <property name="hibernate.url" value="${jdbc.url}" />  
    <property name="hibernate.username" value="${jdbc.username}" />  
    <property name="hibernate.password" value="${jdbc.password}" />
```

```
    ...
```

```
    ...
```

```
    <property name="hibernate.dialect" value="org.hibernate.dialect.MySQL5Dialect" />  
    <property name="hibernate.show_sql" value="true" />  
    <property name="hibernate.format_sql" value="update" />  
    <property name="hibernate.charset" value="UTF-8" />  
    <property name="hibernate.use_sql_comments" value="true" />  
    <property name="hibernate.connection.url" value="${jdbc.url}" />
```

```
com.mysql.jdbc.Driver
5
50
120
100
120
2
true
```

```
GET
HEAD
POST
```

配置文件比较乱，是早起配置的一个spring+hibernate+ibatis的框架，在上下文环境中，看到涉及到roxy的配置项是：

因为找不到其他的解决办法，那么就关于此处的配置升级到spring4.0的配置，看是否能解决此问题
以下是修改后的配置文件（略作整理）：

```
classpath*:jdbc.properties
```

...

...

```
org.hibernate.dialect.MySQL5Dialect  
false  
update  
UTF-8  
true  
{jdbc.url}  
com.mysql.jdbc.Driver  
5
```

50

120

100

120

2

true

```
${jdbc.driverClassName}  
${jdbc.url}  
${jdbc.username}  
${jdbc.password}
```

classpath:sqlConfig.xml

GET
HEAD
POST

新的配置文件中，对事务的管理部分，使用了tx,aop来管理，并且取消了annotation方式的事务管理问题得到解决，而具体的原因还未证明，关于annotation注解：

@Transactional 注解可以被应用于接口定义和接口方法、类定义和类的 public 方法上。

Spring团队的建议是你在具体的类（或类的方法）上使用 @Transactional 注解，而不要使用在类所实现的任何接口上。

```
@Service
@Transactional(rollbackFor=Exception.class) //对当前类的所有方法起作用
@SuppressWarnings("serial")
public class ButtonBo extends GlobalBo {
    ....
    @Transactional(propagation = Propagation.NOT_SUPPORTED, readOnly = true) //具体方法上
    public Button findButton(String buttonid) throws BaseException {
        return hibernateEntityDao.get(Button.class, buttonid);
    }
}
```