

良工说技术

从事java工作近10年，分享有关java相关的知识。

博客园

首页

新随笔

联系

订阅

管理

随笔 - 166

文章 - 1

评论 - 190

阅读 - 123万

公告

欢迎关注公众号：良工说技术



昵称： 良工说技术
园龄： 6年9个月
粉丝： 125
关注： 13
[+加关注](#)

搜索

找找看

谷歌搜索

最新随笔

- 1.我的十年程序员生涯--二战在左，工作在右，很难的选择吗？
- 2.我的十年程序员生涯--考研失利，倒也还好
- 3.我的十年程序员生涯--开篇
- 4.为了安装alien，我更新了yum源，结果还是没装上
- 5.周末折腾了两天，踩了无数个坑，终于把win7装成了centos7
- 6.总算给女盆友讲明白了，如何使用stream流的filter()操作
- 7.这么简单，还不会使用java8 stream流的map()方法吗？
- 8.使springAOP生效不一定要加@EnableAspectJAutoProxy注解
- 9.springAOP和AspectJ有关系吗？如何使用springAOP面向切面编程
- 10.java日常开发必备：list的四种遍历

随笔档案

- 2023年3月(3)
- 2023年2月(2)
- 2022年12月(2)
- 2022年8月(4)

使springAOP生效不一定要加@EnableAspectJAutoProxy注解

在上篇文章《springAOP和AspectJ有关系吗？如何使用springAOP面向切面编程》中遗留了一个问题，那就是在springboot中使用springAOP需要加@EnableAspectJAutoProxy注解吗，网上很多都说需要加这个注解，但是有些情况却是不需要加，就比如我下面的例子，这是为什么，难道网上都说错了吗？

一、效果演示

以下面的例子演示，

业务类，

UserService.java

JAVA 复制 全屏

```
package com.my.template.service;

import com.my.template.entity.User;
import org.springframework.stereotype.Service;

/**
 * @date 2022/8/9 15:28
 */
@Service
public class UserService implements Us{
    @Override
    public void saveUser(User user){
        System.out.println("保存user对象到数据库:"+user);
    }
}
```

切面类，

LogAspect.java

```
package com.my.template.aop;

import org.aspectj.lang.JoinPoint;
import org.aspectj.lang.annotation.AfterReturning;
import org.aspectj.lang.annotation.Aspect;
import org.aspectj.lang.annotation.Before;
import org.aspectj.lang.annotation.Pointcut;
import org.springframework.stereotype.Component;

/**
 * @date 2022/8/11 14:12
 */
@Component
@Aspect
public class LogAspect {
    @Pointcut("execution(* com.my.template.service.UserService.*(..))")
    public void pointCut(){

    }
    @Before(value = "pointCut()")
    public void before(JoinPoint joinPoint){
        System.out.println("方法执行前-20220816");
    }

    @AfterReturning(value = "pointCut()")
    public void after(JoinPoint joinPoint){
        System.out.println("方法执行后-20220816");
    }
}
```

测试的controller

UserController.java

```
package com.my.template.controller;
```

2022年7月(1)
2022年6月(5)
2022年5月(11)
2022年4月(6)
2022年3月(1)
2022年2月(1)
2022年1月(1)
2021年11月(1)
2021年10月(1)
2021年9月(1)
2021年8月(3)
更多

阅读排行榜

- 1. JSON字符串和java对象的互转【js on-lib】(535366)
- 2. tomcat之一： 指定tomcat运行时JDK版本(119060)
- 3. maven之一： maven安装和eclipse集成(117601)
- 4. java线程池之一： 创建线程池的方法(42375)
- 5. mybatis中#和\$符号的区别(37773)

评论排行榜

- 1. 一个程序员的2021总结(28)
- 2. 周末折腾了两天， 踩了无数个坑，终于把win7装成了centos7(20)
- 3. 我的十年程序员生涯--考研失利，倒也还好(8)
- 4. 工作以后难道就不需要继续学习了吗(8)
- 5. JSON字符串和java对象的互转【js on-lib】(8)

最新评论

- 1. Re:我的十年程序员生涯--考研失利，倒也还好
@成佛在西天 感谢评论 苏州大学是211吧，大学挺好得。既然没上岸，那就下海捕鱼。哈哈...
--良工说技术
- 2. Re:我的十年程序员生涯--考研失利，倒也还好
苏州大学，没上岸
--成佛在西天
- 3. Re:我的十年程序员生涯--考研失利，倒也还好
@myfish3 感谢评论 是08级，下次注意，再次感谢...
--良工说技术
- 4. Re:我的十年程序员生涯--考研失利，倒也还好
这叫08级，不是08届
--myfish3
- 5. Re:我的十年程序员生涯--考研失利，倒也还好
@yi念之间 感谢评论 学历现在愈发重要，尤其是在一线城市。学历不够经验来凑，后面还会提到这方面...
--良工说技术

```
import com.my.template.entity.User;
import com.my.template.service.Us;
import com.my.template.service.UserService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

/**
 * @date 2022/8/9 15:35
 */
@RestController
public class UserController {
    @Autowired
    private Us us;
    @RequestMapping("/saveUser")
    public String saveUser() {

        User user=new User();
        user.setId("1");
        user.setName("张三");
        us.saveUser(user);
        return "success";
    }
}
```

sprinboot的启动类，

BootServer.java

```
package com.my.template;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.EnableAspectJAutoProxy;
import org.springframework.context.annotation.ImportResource;

/**
 * 启动类
 * @date 2022/6/3 21:32
 */
@SpringBootApplication()
public class BootServer {
    public static void main(String[] args) {
        try {
            SpringApplication.run(BootServer.class);
        }catch (Exception e){
            e.printStackTrace();
        }
    }
}
```

测试结果如下，

```
2022-08-16 22:30:44.082 INFO 25716 --- [nio-9099-exec-1] o.s.web.servlet.DispatcherServlet
: Completed initialization in 6 ms
方法执行前-20220816
保存user对象到数据库:User{name='张三', id='1'}
方法执行后-20220816
```

从上面的测试结果来看，没有加@EnableAspectJAutoProxy注解，但是AOP生效了，这是为什么？

二、为什么不加@EnableAspectJAutoProxy切面生效

关于这个问题我排查了很久，最后在依赖中找到了原因，看下pom文件

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

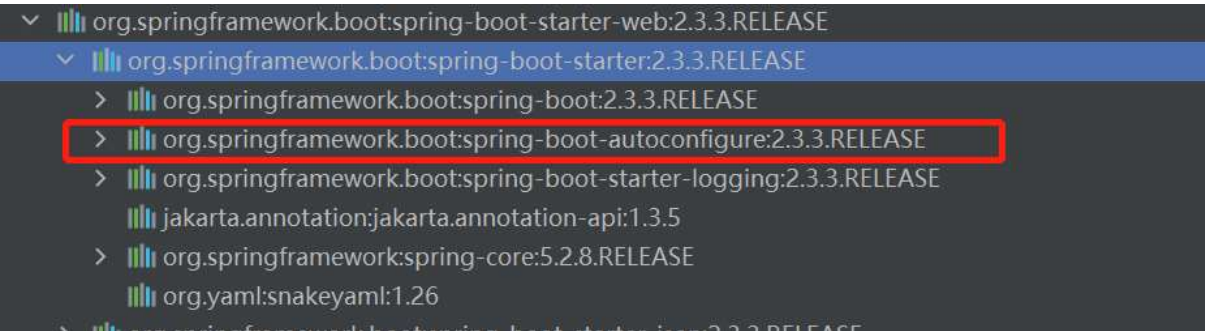
    <groupId>org.example</groupId>
    <artifactId>springTemplate</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>8</maven.compiler.source>
        <maven.compiler.target>8</maven.compiler.target>
    </properties>
```

```
<dependencies>
    <!--spring-boot的web依赖-->
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
        <version>2.3.3.RELEASE</version>
    </dependency>
    <!--自定义的starter-->
    <dependency>
        <groupId>org.example</groupId>
        <artifactId>customer-spring-boot-starter</artifactId>
        <version>1.0-SNAPSHOT</version>
    </dependency>
    <!--使用springAOP需要引入该依赖-->
    <dependency>
        <groupId>org.aspectj</groupId>
        <artifactId>aspectjweaver</artifactId>
        <version>1.8.13</version>
    </dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
            <executions>
                <execution>
                    <goals>
                        <goal>repackage</goal>
                    </goals>
                </execution>
            </executions>
        </plugin>
    </plugins>
</build>
</project>
```

在依赖中有spring-boot-starter-web的依赖，该依赖有下面的依赖，



会引入spring-boot-autoconfigure的依赖，这是自动装配的依赖，也就是会读取其下的spring.factories文件，在该文件中有下面的配置，

```
# Auto Configure
org.springframework.boot.autoconfigure.EnableAutoConfiguration=\
org.springframework.boot.autoconfigure.admin.SpringApplicationAdminJmxAutoConfiguration,\
org.springframework.boot.autoconfigure.aop.AopAutoConfiguration,\
org.springframework.boot.autoconfigure.amqp.RabbitAutoConfiguration,\
org.springframework.boot.autoconfigure.batch.BatchAutoConfiguration,\
```

没错就是因为AopAutoConfiguration类的问题。下面看具体原因。

三、原因分析

要看具体原因，我们就要打开AopAutoConfiguration这个类看下，

```
/**
 * {@link org.springframework.boot.autoconfigure.EnableAutoConfiguration}
 * Auto-configuration for Spring's AOP support. Equivalent to enabling
 * {@link EnableAspectJAutoProxy @EnableAspectJAutoProxy} in your configuration.
 * <p>
 * The configuration will not be activated if {@literal spring.aop.auto=false}. The
 * {@literal proxyTargetClass} attribute will be {@literal true}, by default, but can be
 * overridden by specifying {@literal spring.aop.proxy-target-class=false}.
 *
 * @author Dave Syer
 * @author Josh Long
 * @since 1.0.0
 * @see EnableAspectJAutoProxy
 */
@Configuration(proxyBeanMethods = false)
@ConditionalOnProperty(prefix = "spring.aop", name = "auto", havingValue = "true", matchIfMissing = true)
public class AopAutoConfiguration {
```

先看注释吧，注释中说**AopAutoConfiguration**等同于**@EnableAspectJAutoProxy**注解，也就是该类起的作用和**@EnableAspectJAutoProxy**是一样的，再看该类上的注解，重点看**@ConditionalOProperty**

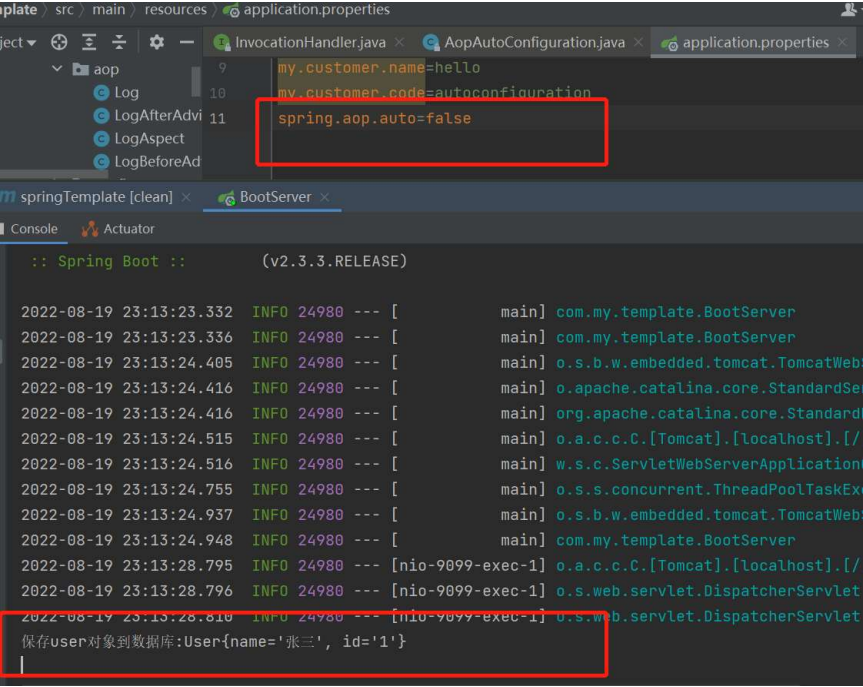
注解中的内容，意思是如果在配置文件中” spring.aop.auto “的配置，如果不配置为true，否则可以配置为false，现在我的配置文件中是没有该配置项的，

```
server.port=9099
spring.datasource.type=com.mysql.cj.jdbc.MysqlDataSource
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource..driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.username=yh_dev
spring.datasource..password=DvpJe2x
spring.datasource.url=jdbc:mysql://10.0.0.37:3306/channel_center
#?????
my.customer.name=hello
my.customer.code=autoconfiguration
```

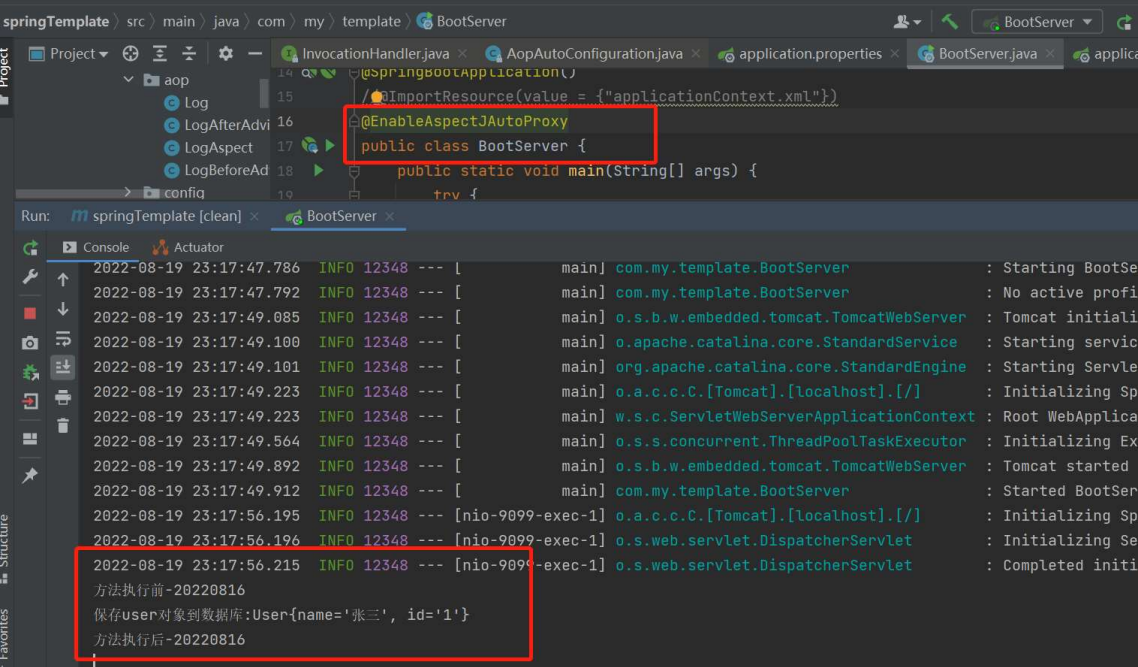
那么我现在增加该配置，并设置为false，

```
server.port=9099
spring.datasource.type=com.mysql.cj.jdbc.MysqlDataSource
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource..driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.username=yh_dev
spring.datasource..password=DvpJe2x
spring.datasource.url=jdbc:mysql://10.0.0.37:3306/channel_center
#?????
my.customer.name=hello
my.customer.code=autoconfiguration
spring.aop.auto=false
```

重启服务之后，看测试结果，



从测试结果可以看到springAOP没有起作用，现在在启动类上加上@EnableAspectJAutoProxy注解，看下测试结果，



从上面的测试结果可以看到，添加了@EnableAspectJAutoProxy注解springAOP生效了。

综上，在springboot环境下，由于存在spring-boot-autoconfigure依赖，默认会注入AopAutoConfiguration配置类，该类的作用等同于@EnableAspectJAutoProxy注解，所以在这种情况下可以不加@EnableAspectJAutoProxy注解，AopAutoConfiguration可以通过spring.aop.auto属性控制；

四、总结

本文主要分析了在springboot环境下，不加@EnableAspectJAutoProxy注解springAOP仍然生效的问题。**为了保险期间请一律加上@EnableAspetJAutoProxy注解。**

1. AopAutoConfiguration类等同于@EnableAspectJAutoProxy注解;

2. spring.aop.auto=ture/false属性可以控制AopAutoConfiguration类是否生效;



感谢分享、关注、收藏。公众号：【良工说技术】

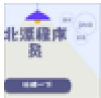
分类: [springboot](#)

标签: [springboot](#), [springboot源码](#), [AOP](#)

好文要顶

关注我

收藏该文



良工说技术

粉丝 - 125 关注 - 13

[+加关注](#)

« 上一篇: [springAOP和AspectJ有关系吗？如何使用springAOP面向切面编程](#)

» 下一篇: [这么简单，还不会使用java8 stream流的map\(\)方法吗？](#)

posted @ 2022-08-20 10:10 良工说技术 阅读(729) 评论(2) 编辑 收藏 举报

[刷新评论](#) [刷新页面](#) [返回顶部](#)

登录后才能查看或发表评论，立即 [登录](#) 或者 [逛逛](#) 博客园首页

- 【推荐】[博客园人才出海服务](#)第一站，联合日本好融社推出日本IT人才移民直通车
- 【推荐】[中国云计算领导者](#)：阿里云轻量应用服务器2核2G，新用户低至108元/年



编辑推荐：

- [实现和 CSS 一样的 easing 动画？直接看 Mozilla、Chromium 源码](#)

· [80 张图带你一步一步推演 slab 内存池的设计与实现](#)

· [记一次 .NET 某手术室行为信息系统 内存泄露分析](#)

· [前端性能精进 —— 构建](#)

· [Windows 服务器高物理内存占用问题排察](#)

即构专区：

- [《Among Us》火爆全球，实时语音助力派对游戏开启第二春](#)

· [技术干货：在线教育平台如何保障课件数据安全](#)

· [【活动回顾】泛娱乐社交行业“闪聊”新玩法](#)

· [即视角 | 泛娱乐场景下的银发经济：老龄用户画像及触网行为分析](#)

· [小小的耳返功能，为什么你就是实现不了？](#)

历史上的今天：

2016-08-20 JSON字符串和java对象的互转【json-lib】