

☰ 01、SpringAop的有没有应用和使用过

在上一个公司的我工作的时候，大部分的项目都是以Spring架构项目为主，在平时开发项目的时候，在xxx项目其实也使用AOP去完成的一些业务，也利用AOP去解决了一些业务问题，其中有两个记忆比较深刻：

- 1：使用AOP完成接口的性能监控：使用AOP的目的：AOP可以降低代码耦合度，以及解决和快速的排查项目中存在的性能隐患。具体实现的方式是：通过日志进行监控，并且结合ELK进行数据存储和展示，完成的接口的性能的分析。
- 2：使用AOP完成业务限流工作：我记得很清楚的是，我们在做xxx模块（抢购，下单，注册，秒杀）的时候，刚开始网站用户量不是特别的大，没什么太大问题，但是随着项目和业务的推荐，用户量也在上升，有反馈出现了重复下单和代码攻击的等出现问题，这个使用限流+锁的机制来完成的这种并发造成的问题。
- 3：对一些业务进行一些安全限制：比如项目中存在一些需要权限的操作和控制接口。比如项目一些财务核算，订单查询，也自己AOP拦截思想进行开发。

☰ 02、谈谈你对SpringAop认识和理解

级别：框架基础

初衷：看看对框架spring的认识有多深

开场话术1：开发项目的时候，大部分情况我们项目是以spring为主，我对AOP认识和理解是：

开场话术2：在上一个公司的我工作的时候，大部分的项目都是以Spring架构项目为主，在平时开发项目的时候，在xxx项目其实也使用AOP去完成的一些业务，也利用AOP去解决了一些业务问题，我自己对理解是：

AOP它是面向切面的一种编程思想，它主要的目的就可以把：对象执行方法的过程中，进行一些业务逻辑代码的织入。这中机制可以大大提高代码的质量，从而达到高内聚低耦合的目的，同时让我们业务和业务直接的耦合性降低，维护的效率更高。AOP本身它是值是一种思想，而Spring的Aop的实现，依托于JDK本身提供的jdk动态代理和CGLIB代理。那么Spring的Aop对这种两种代理方式都进行了支持。

（1）在平时开发代码的时候，完全可以把一些常见的，常用的功能代码进行封装，尽量做到动态配置。不同的功能模块只需要进行织入即可。

（2）定义业务逻辑的模板，比如说如果要解决某一个业务功能，如果页面类似，可以按照基本的框架进行组合，然后使用配置平台进行可控化配置即可。

03、你谈谈你对SpringAop核心概念认识是什么？

首先springaop实现是通过：jdk动态代理和cglib代理实现的。

无论是那种代理方式其实首先必须要有几个核心要素：

- 代理类
- 代理对象
- 目的对象
- 接口（cglib接口不需要定义）

但是传统的代理方式它的粒度比较大。以及自我实现的控制业务的能力都需要自己去开发。然而spring框架对动态代理进行了一系列的增强处理。比如增加：

- 切面
 - Aop具体做事的地方，它连接点和通知聚焦点。所有增强的业务在这里统一编排和完成。
 - @Aspect
 - @Component
- 连接点
 - 每个对象中的每个方法都是连接点
- 切入点
 - 方法执行增强逻辑的条件判断
- 通知
 - 满足切入点表达的是的具体织入的逻辑
- 织入：其实你说这两个点，其实都是spring框架自身来完成的，织入它是动作，就是指是把切面应用到切点对应的连接点的过程。切面在指定连接点被织入到目标对象中。（spring框架做的事情）

一个代理对象执行目标方法的时候，如果执行的方法满足切点表达式，就会进入到具体通知中，并且所有的过程全部在切面完成。我们把这个所有的过程称之为：织入

SpringAop和ioc关系什么？

springioc是spring基础，它管理了项目中所有满足的bean，并且放入到容器中。而springaop的基础是ioc。没有ioc，是不复存在的，原因是：springaop底层是通过jdk动态代理和cglib代理来完成的。不论是用那种代理方式，它都必须要有具体代理的目标对象，而目标对象从何而来，没错就是通过springioc容器而来。

☰ 通过最近几天的学习AOP

- 开发会清晰一点。也知道怎么去应用
 - 提供了一个方向：你可以查看spring源码它是如何把ioc的对象转化代理对象？
 - 从哪里开始看：从Spring-Aop.jar看 ----- Spring的生命周期---提供各种接口
 - 从哪里开始看：从Spring-Mybaitis.jar看 ----- Spring的生命周期
-
- 接口有什么用？
 - 谁来实现和扩展
 - spring是怎么加载这个接口----下所有实现的类的？
 - 怎么放，怎么取

核心：java面向对象，执行方法，给属性赋值。