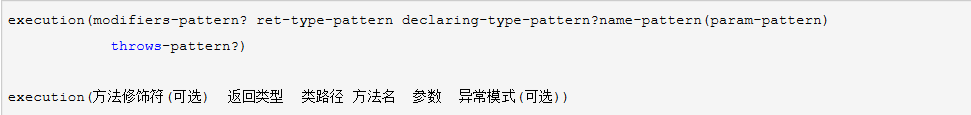
### 内容介绍

1. 什么是AOP
2. AOP有什么用
3. AOP的应用场景
4. 一些AOP的常用的知识点
5. 在面试中常常AOP被常常被问到的知识点

### Spring中AOP

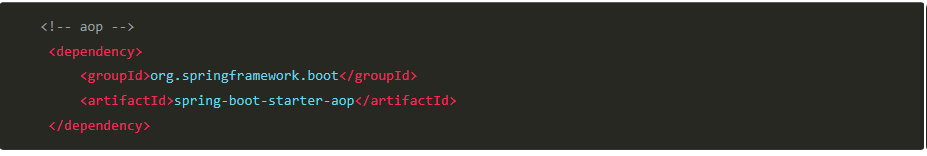
1. 什么是AOP
   1. 面向切面编程,它是一个概念，一个规范，本身并没有设定具体语言的实现
2. AOP有什么用
   1. 在预编译期间和运行期间动态代理实现不改变源代码给程序统一增加功能的一种技术
3. AOP的应用场景
   1. 权限
   2. 缓存
   3. 异常处理
   4. log记录
4. 一些AOP的常用的知识点
   1. 使用@Aspect和声明一个切面类（这个@Aspect注释告诉Spring这是个切面类）
   2. 使用@PointCut声明一个切入点例如@PointCut(...........)
      1. execution: 匹配连接点
      2. within: 某个类里面
      3. this: 指定AOP代理类的类型
      4. target:指定目标对象的类型
      5. args: 指定参数的类型
      6. bean:指定特定的bean名称，可以使用通配符（Spring自带的）
      7. @target： 带有指定注解的类型
      8. @args: 指定运行时传的参数带有指定的注解
      9. @within: 匹配使用指定注解的类
      10. @annotation:指定方法所应用的注解
   3. 这里我们主要介绍execution的匹配方法，因为大多数时候都会用这个来定义pointcut：



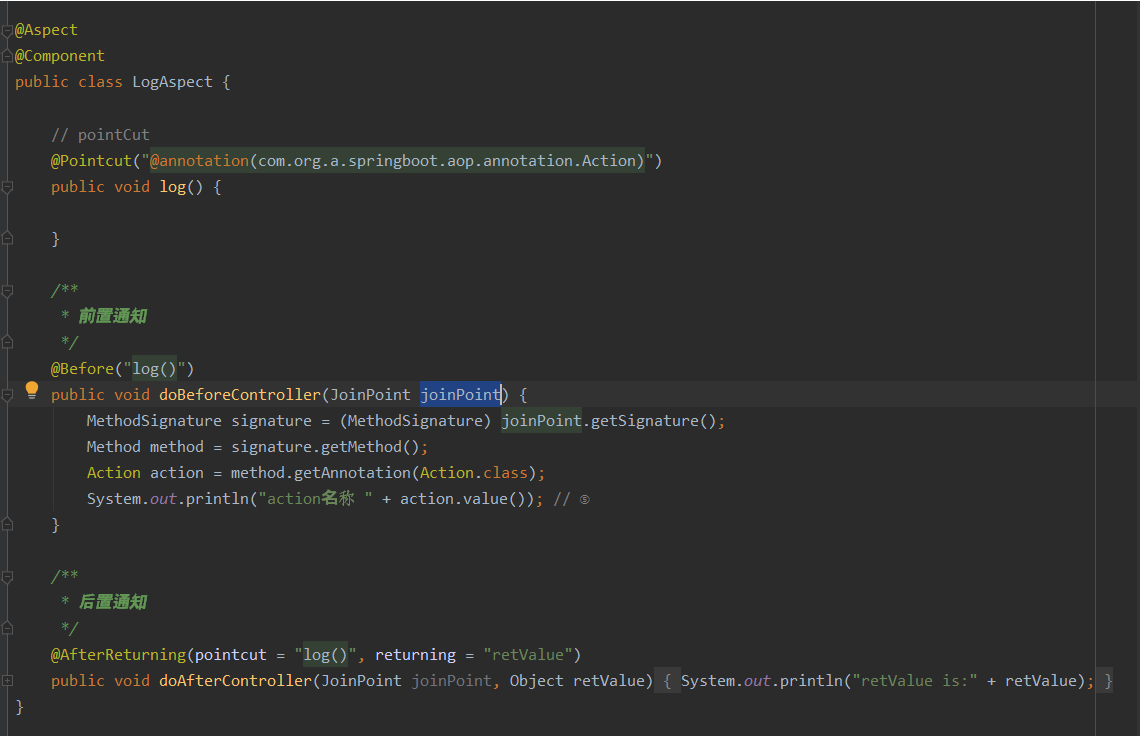
比如@Pointcut("execution(public \* com.feishenzhihoudewo.controller..\*.\*(..))")

* 1. 声明建议
     1. @Before ————在切入点的方法运行之前要执行的
     2. @After————这个注解就是在切入的方法运行完之后把我们的advice增强加进去
     3. @AfterReturning————在切入点的方法返回时执行
     4. @AfterThrowing—————在切入点的方法抛出异常时执行
     5. @Around————在切入点的方法运行之前和之后执行

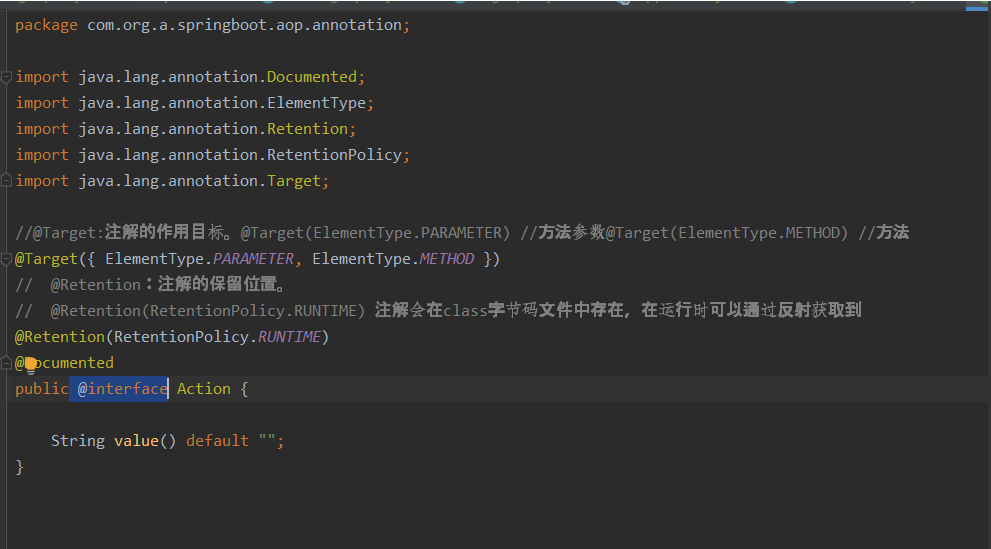
1. 在SpringBoot中搭建AOP
   1. 添加依赖



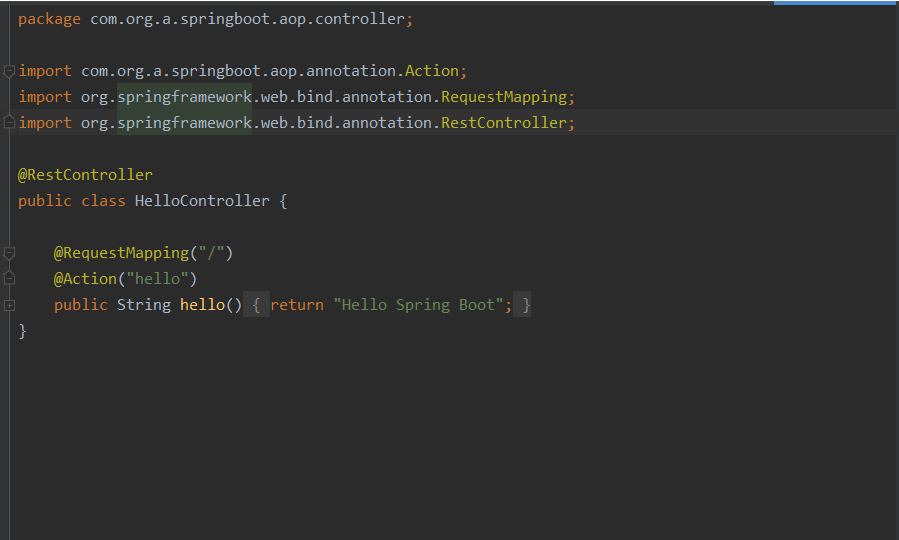
* 1. 编写切面类



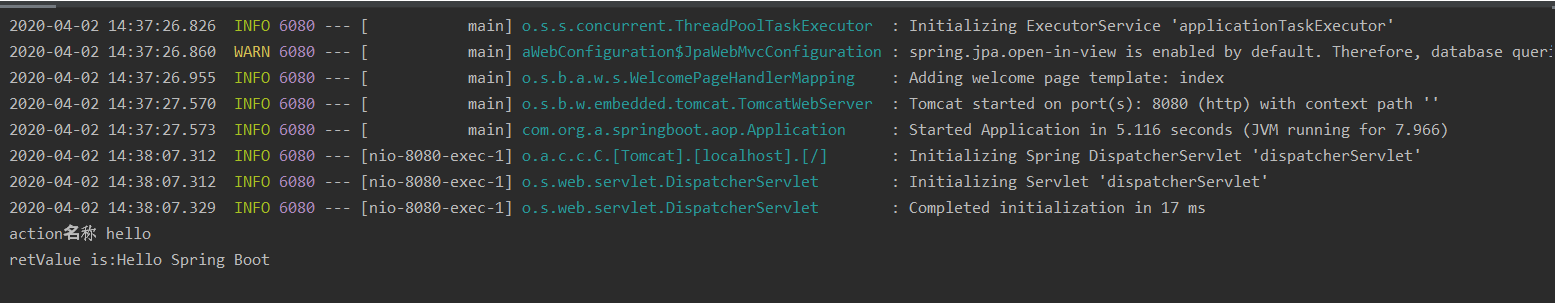
* 1. 编写拦截规则的注解



* 1. 编写controller并将注解加入



* 1. 启动程序，访问[http://localhost:8080/](http://localhost:8080/" \t "http://blog.longjiazuo.com/archives/_blank)，看到控制台打印如下日志



* 1. 程序地址https://github.com/baoqintian/springBootDemo