#### Spring Boot

##### 之面向切面编程（AOP）

问题描述

Spring Boot 2.2.5

JDK 8

一些看上去不可思议的特性往往与JVM有着紧密的联系，比如重量级锁synchronized、又如这边笔记中提及的面向切面编程AOP

在Spring Boot项目中使用AOP功能的配置是“老三样儿”：

* 添加Maven依赖
* 通过注解开启功能
* 定义切面

解决

1. 添加Maven依赖

<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter</artifactId>  
 <version>2.3.0.RELEASE</version>  
</dependency>

<dependency>  
 <groupId>org.aspectj</groupId>  
 <artifactId>aspectjweaver</artifactId>  
 <version>1.9.5</version>  
</dependency>

1. 通过注解开启功能

@Configuration  
@EnableAspectJAutoProxy  
@ComponentScan  
public class ApplicationConfig {  
}

1. 定义切面

@Aspect  
@Component  
public class LogExecutor {  
 @Around("execution(public void com.bunny.aop.worker.Pot.feedAnimal())")  
 public Object log(ProceedingJoinPoint proceedingJoinPoint) throws Throwable {  
 System.*out*.println("before feed animal");  
 return proceedingJoinPoint.proceed();  
 }  
}

笔记

1. 切面类需要注册成Java Bean
2. AspectJ’s dynamic crosscutting support comes in the form of advice. Advice is the code executed at a join point selected by a pointcut. Advice can execute before, after, or around the join point. The body of advice is much like a method body—it encapsulates the logic to be executed upon reaching a join point.
3. Spring AOP应用场景
   1. 打印日志
   2. 调优：打印方法执行时间
   3. 鉴权

□