首先在你的Maven的pom文件里加入aop的依赖：

<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-aop</artifactId>  
 </dependency>

在spring boot里面一切配置都是很简单的，下面为我所有被请求到的controller加上Aop的功能吧，看码：

package me.jiaobuchong.admin.config;  
  
**import** javax.servlet.http.HttpServletRequest;  
  
**import** org.aspectj.lang.ProceedingJoinPoint;  
**import** org.aspectj.lang.annotation.Around;  
**import** org.aspectj.lang.annotation.Aspect;  
**import** org.aspectj.lang.annotation.Pointcut;  
**import** org.springframework.context.annotation.Configuration;  
**import** org.springframework.web.context.request.RequestAttributes;  
**import** org.springframework.web.context.request.RequestContextHolder;  
**import** org.springframework.web.context.request.ServletRequestAttributes;  
**import** com.google.gson.Gson;  
**import** org.slf4j.Logger;  
**import** org.slf4j.LoggerFactory;;  
  
*/\*\*  
 \* Created by jiaobuchong on 12/23/15.  
 \*/*  
@**Aspect** *//定义一个切面*  
@**Configuration**  
**public** **class** **LogRecordAspect** {  
**private** **static** **final** **Logger** logger = **LoggerFactory**.getLogger(**UserInterceptor**.**class**);  
  
 *// 定义切点Pointcut*  
 @**Pointcut**("execution(\* com.jiaobuchong.web.\*Controller.\*(..))")  
 **public** void excudeService() {  
 }  
  
 @**Around**("excudeService()")  
 **public** **Object** doAround(**ProceedingJoinPoint** pjp) **throws** **Throwable** {  
 **RequestAttributes** ra = **RequestContextHolder**.getRequestAttributes();  
 **ServletRequestAttributes** sra = (**ServletRequestAttributes**) ra;  
 **HttpServletRequest** request = sra.getRequest();  
  
 **String** url = request.getRequestURL().toString();  
 **String** method = request.getMethod();  
 **String** uri = request.getRequestURI();  
 **String** queryString = request.getQueryString();  
 logger.info("请求开始, 各个参数, url: {}, method: {}, uri: {}, params: {}", url, method, uri, queryString);  
  
 *// result的值就是被拦截方法的返回值*  
 **Object** result = pjp.proceed();  
 **Gson** gson = new **Gson**();  
 logger.info("请求结束，controller的返回值是 " + gson.toJson(result));  
 **return** result;  
 }  
}

只要加上上面这个类，Aop就算配置好了，@Configuration这个Annotation就是JavaConfig的典型代表，Spring boot在启动时会会自动去加载这些配置，实现相应的配置功能