



链滴

# Spring Aop 获取入参和出参

作者: [boolean-dev](#)

原文链接: <https://ld246.com/article/1609852542166>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# Spring Aop 获取入参和出参

## 1. 概述

本次，我利用 Spring Aop 的注解方式获取切入点的入参和出参，因为比较简单，所以就直接上代码。

## 2. 代码编写

### 注解类

```
/**
 * 日志注解
 *
 * @author Jiantao Yan
 * @title: Log
 * @date 2021/1/5 20:33
 */
@Target({ElementType.METHOD})
@Retention(RetentionPolicy.RUNTIME)
public @interface Log {
}
```

### 切入点

```
/**
 * AOP 切入点测试
 *
 * @author Jiantao Yan
 * @title: AopService
 * @date 2021/1/5 20:29
 */
@Slf4j
@Service
public class AopService {

    @Log
    public User save(User user) {
        log.info("----->save user={}", user);
        return user;
    }
}
```

### 切面

```
/**
 * 日志切面
 *
 * @author Jiantao Yan
```

```

* @title: LogAspect
* @date 2021/1/5 20:33
*/
@Slf4j
@Aspect
@Component
public class LogAspect {

    /**
     * 以注解为切入点
     */
    @Pointcut("@annotation(com.tao.learn.demo.aop.Log)")
    public void pointcut() {
    }

    /**
     * 环绕通知
     * @param proceedingJoinPoint
     */
    @SneakyThrows
    @Around("pointcut()")
    public void around(ProceedingJoinPoint proceedingJoinPoint) {
        // 获取入参
        Object[] args = proceedingJoinPoint.getArgs();
        for (Object arg : args) {
            // 入参是否为 user
            if (arg instanceof User) {
                log.info("----->入参为user={}", (User)arg);
            }
            log.info("----->入参 arg={}", arg);
        }

        // 获取出参
        Object result = proceedingJoinPoint.proceed();
        // 出参是否为 user
        if (result instanceof User) {
            log.info("----->入参为user,result={}", (User)result);
        }
        log.info("----->出参.result={}", result);
    }
}

```

## 实体类

```

/**
 * 用户
 *
 * @author Jiantao Yan
 * @title: User
 * @date 2021/1/5 20:30
 */
@Data
public class User {

```

```
private Long id;

private String username;

private Integer age;
}
```

### 3. 结果验证

#### 单元测试编写

```
@Slf4j
@SpringBootTest
class AopServiceTest {

    @Autowired
    private AopService aopService;

    @Test
    void save() {
        User user = new User();
        user.setId(1L);
        user.setAge(19);
        user.setUsername("booleandev");
        User user1 = aopService.save(user);
        log.info("----->user1={}", user1);
    }
}
```

#### 结果

```
2021-01-05 21:01:13.759 INFO 7440 --- [      main] com.tao.learnrn.demo.aop.LogAspect
: ----->入参为user=User(id=1, username=booleandev, age=19)
2021-01-05 21:01:13.774 INFO 7440 --- [      main] com.tao.learnrn.demo.aop.LogAspect
: ----->入参 arg=User(id=1, username=booleandev, age=19)
2021-01-05 21:01:13.786 INFO 7440 --- [      main] com.tao.learnrn.demo.aop.AopService
: ----->save user=User(id=1, username=booleandev, age=19)
2021-01-05 21:01:13.787 INFO 7440 --- [      main] com.tao.learnrn.demo.aop.LogAspect
: ----->入参为user,result=User(id=1, username=booleandev, age=19)
2021-01-05 21:01:13.787 INFO 7440 --- [      main] com.tao.learnrn.demo.aop.LogAspect
: ----->出参.result=User(id=1, username=booleandev, age=19)
2021-01-05 21:01:13.787 INFO 7440 --- [      main] com.tao.learnrn.demo.aop.AopServiceT
st : ----->user1=null
```