

# 日志处理技术方案

## 项目需求

- 1. 为smpe-admin框架添加日志处理模块
- 2. 持久化存储日志处理
- 3. 后台图形化界面管理
- 4. 具有可扩展性方便后期改造升级

## 技术选型

### Spring框架AOP

实现smpe-admin框架的日志处理功能，采用的是Spring框架中的AOP功能。AOP 要达到的效果是，保证开发者不修改源代码的前提下，去为系统中的业务组件添加某种通用功能。

按照 AOP 框架修改源代码的时机，可以将其分为两类：

- 静态 AOP 实现， AOP 框架在编译阶段对程序源代码进行修改，生成了静态的 AOP 代理类（生成的 \*.class 文件已经被改掉了，需要使用特定的编译器），比如 AspectJ。
- 动态 AOP 实现， AOP 框架在运行阶段对动态生成代理对象（在内存中以JDK 动态代理，或 CGLib 动态地生成 AOP 代理类），如 SpringAOP。

下面给出常用 AOP 实现比较：

类别	机制	原理	优点	缺点
静态 AOP	静态织入	在编译起，切面直接以字节码的形式编译到目标字节码文件中	对系统无性能影响	灵活性不够
动态 AOP	JDK 动态代理	在运行期，目标类加载后，为接口动态生成代理类，将切面织入到代理类中	相对于静态 AOP 更加灵活	切入的关注点需要实现接口。对系统有一点性能影响
动态字节码生成	CGLIB	在运行期，目标类加载后，动态生成目标类的子类，将切面逻辑加入到子类中	没有接口也可以织入	扩展类的实例方法用 final 修饰时，则无法进行织入
自定义类加载器		在运行期，目标类加载前，将切面逻辑加到目标字节码里	可以对绝大部分类进行织入	代码中如果使用了其他类加载器，则这些类将不会织入
字节码转换		在运行期，所有类加载器加载字节码前进行拦截	可以对所有类进行织入	

# 为什么选用AOP

想象下面的场景，开发中在多个模块间有某段重复的代码，我们通常是怎么处理的？

在传统的面向过程编程中，我们也会将这段代码，抽象成一个方法，然后在需要的地方分别调用这个方法，这样当这段代码需要修改时，我们只需要改变这个方法就可以了。然而需求总是变化的，有一天，新增了一个需求，需要再多出做修改，我们需要再抽象出一个方法，然后再在需要的地方分别调用这个方法，又或者我们不需要这个方法了，我们还是得删除掉每一处调用该方法的地方。实际上涉及到多个地方具有相同的修改的问题我们都可以通过 AOP 来解决。

## Aop的基本原理

### 核心元素

#### 1、Pointcut-切入点表达式

Spring框架中的org.springframework.aop.Pointcut接口是一个重要接口，能够用于对不同的类以及方法设置不同的通知，即对目标对象的方法通过相同的切入点设置不同的增强。Spring AOP 所支持的 AspectJ 切点指示器。

表 4.1 Spring 借助 AspectJ 的切点表达式语言来定义 Spring 切面

AspectJ 指示器	描述
arg()	限制连接点匹配参数为指定类型的执行方法
@args()	限制连接点匹配参数由指定注解标注的执行方法
execution()	用于匹配是连接点的执行方法
this()	限制连接点匹配 AOP 代理的 Bean 引用为指定类型的类
target()	限制连接点匹配目标对象为指定类型的类
@target()	限制连接点匹配特定的执行对象，这些对象对应的类要具备指定类型的注解
within()	限制连接点匹配指定的类型
@within()	限制连接点匹配指定注解所标注的类型（当使用 Spring AOP 时，方法定义在由指定的注解所标注的类里）
@annotation	限制匹配带有指定注解连接点

当我们查看上面展示的这些spring支持的指示器时，注意只有execution指示器是唯一的执行匹配，而其他的指示器都是用于限制匹配的。这说明execution指示器是我们在编写切点定义时最主要使用的指示器，在此基础上，我们使用其他指示器来限制所匹配的切点。

多个匹配之间我们可以使用链接符 `&&`、`||`、`!` 来表示“且”、“或”、“非”的关系。

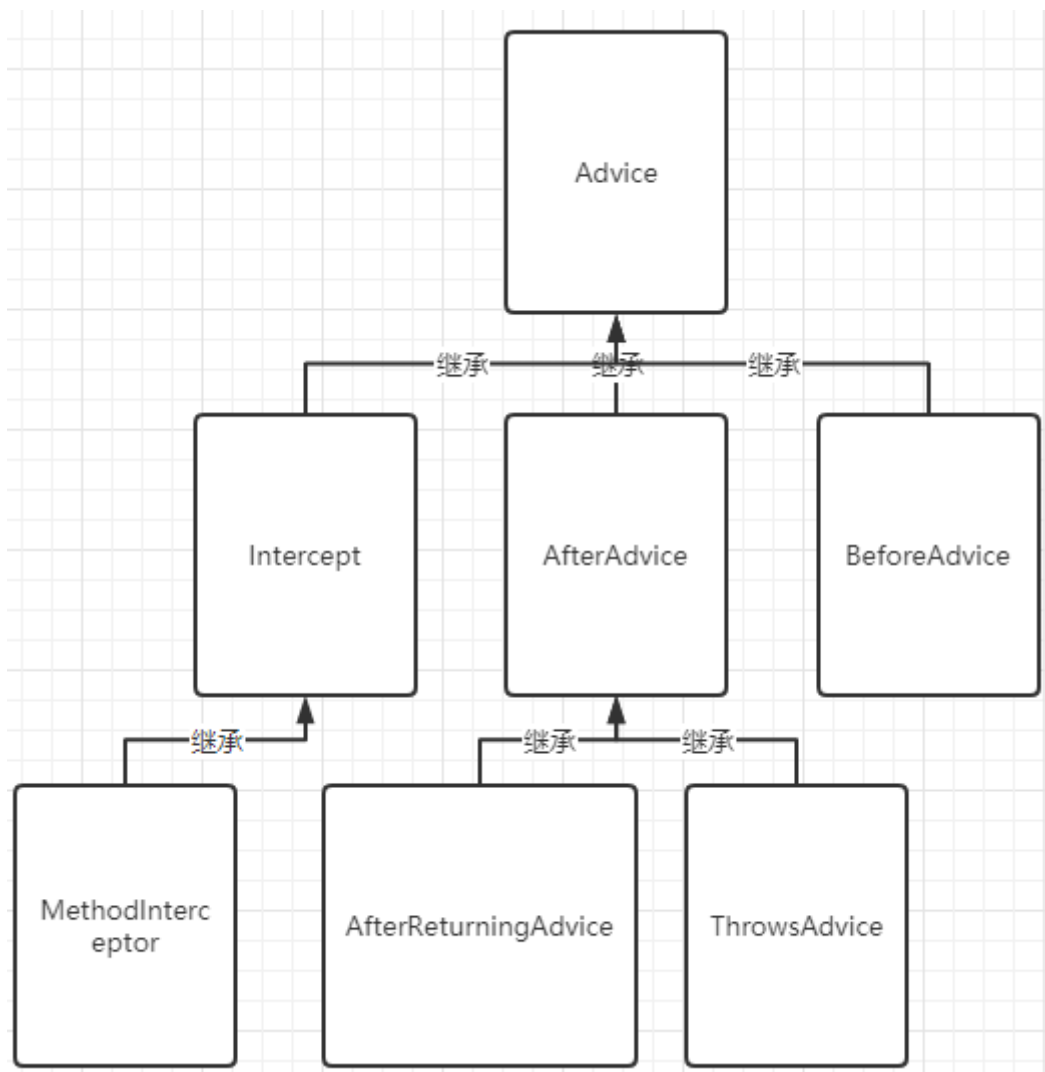
举例：

```
1 //限定该切点仅匹配的包是 com.sharpcj.aopdemo.test1,可以使用
2 execution(* com.sharpcj.aopdemo.test1.IBuy.buy(..)) &&
  within(com.sharpcj.aopdemo.test1.*)
```

#### 2、Advice-通知

Spring框架提供几个常用的Advice，如：Before Advice、Throws Advice、Throws Advice、After Returning Advice等.....但是你可以通过实现Advice类型的接口来扩展你想要的Advice。通知与切入点表达式相关联，它与切入点配合用于完成对目标方法的增强，pointcut负责匹配目标对象的目标方法，advice主要负责通知。

Advice继承图：



常用的advice包括5种，分别是

- BeforeAdvice

这个类型的Advice会在方法执行之前调用。不能够改变返回值，不能够改变目标方法的流程，也不能中断流程的处理过程（除非抛出异常）

- AfterAdvice

在方法调用之后，做处理。可以改变返回值，不能够改变目标方法的流程，也不能中断流程的处理过程（除非抛出异常）

- AfterReturningAdvice

在方法调用之后，做处理。不能够改变返回值，不能够改变目标方法的流程，也不能中断流程的处理过程（除非抛出异常）

- MethodInterceptor

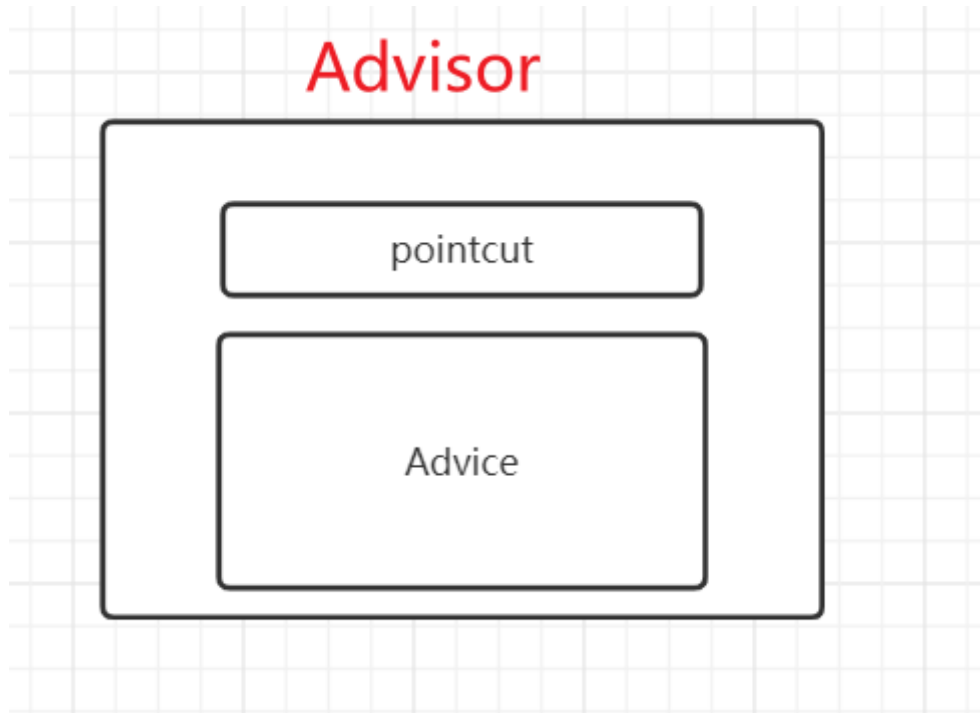
在方法调用之前以及之后，做处理。可以改变返回值，也可以改变流程。

- ThrowsAdvice

在方法抛出异常后，做处理。当该通知处理完异常后，会简单地将异常再次抛出给目标调用方法。

### 3、Advisor-增强

在Spring框架中一个Advisor就是一个切面，它包含一个Advice和相关联的Pointcut表达式。



## 自定义日志处理AOP

### 创建接口日志记录的注解 @Log

```
1 @Target(ElementType.METHOD)
2 @Retention(RetentionPolicy.RUNTIME)
3 public @interface Log {
4     String value() default "";
5 }
```

### 通过注解配置 Spring AOP

```
1 @Component
2 @Aspect
3 public class LogAspect {
4     private final ISysLogService sysLogService;
5
6     @Pointcut("@annotation(marchsoft.annotation.Log)")
7     public void logPointcut() {
8         // 该方法无方法体,主要为了让同类中其他方法使用此切入点
9     }
10
11     @Around("logPointcut()")
12     public Object logAround(ProceedingJoinPoint joinPoint) throws Throwable
13     {
14         result = joinPoint.proceed();
15         //获取日志信息进行持久化
16         return result;
17     }
18 }
```

```

16     }
17
18     @AfterThrowing(pointcut = "logPointcut()", throwing = "e")
19     public void logAfterThrowing(JoinPoint joinPoint, Throwable e) {
20         //异常处理
21     }
22 }

```

## 持久化存储

### 表设计

- sys\_log表

列名	数据类型	长度	默认	主键?	非空?	Unsigned	自增?	Zerofill?	注释
id	bigint	20		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	日志id
user_id	bigint	20	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	操作用户id
description	varchar	255	''	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	描述
log_type	tinyint	3	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	日志类型
method	varchar	255	''	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	方法名
params	text			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	参数
request_ip	varchar	255	''	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	请求ip
request_time	bigint	20	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	请求耗时(毫秒值)
address	varchar	255	''	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	地址
browser	varchar	255	''	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	浏览器
exception_detail	text			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	详细异常
create_by	bigint	20	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	创建人id
update_by	bigint	20	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	更新人id
create_time	datetime			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	创建时间
update_time	datetime			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	更新时间
is_deleted	bit	1	b'0'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	软删除(默认值为0, 1为1)

- sys\_user表

列名	数据类型	长度	默认	主键?	非空?	Unsigned	自增?	Zerofill?	注释
id	bigint	20		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	ID
dept_id	bigint	20	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	部门id
username	varchar	255	''	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	用户名
nick_name	varchar	255	''	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	昵称
gender	bit	1	b'0'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	性别(0为男默认, 1为女)
phone	varchar	255	''	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	手机号码
email	varchar	255	''	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	邮箱
avatar_path	varchar	255	''	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	头像路径
password	varchar	255	''	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	密码
is_admin	bit	1	b'0'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	是否为admin账号
enabled	bit	1	b'1'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	状态: 1启用(默认)、0禁用
create_by	bigint	20	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	创建者id
update_by	bigint	20	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	更新者id
pwd_reset_time	datetime			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	修改密码的时间
create_time	datetime			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	创建日期
update_time	datetime			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	更新时间
is_deleted	bit	1	b'0'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	软删除(默认值为0, 1为1)

## 项目源码

	后端源码	前端源码
GitHub	<a href="https://github.com/sanyueruanjian/supe-admin.git">https://github.com/sanyueruanjian/supe-admin.git</a>	<a href="https://github.com/sanyueruanjian/smp-e-admin-web.git">https://github.com/sanyueruanjian/smp-e-admin-web.git</a>

编辑人：张玉坤

新建于：2021.2.20

审核人：任世伟