

Please visit website: <http://cxyroad.com>

SpringBoot 实现热插拔 AOP解决日志开关需求！！

=====

大家都知道可以用aop来实现日志管理，但是如何动态的实现日志管理呢？

前置知识

* **Advice:**

> org.aopalliance.aop.Advice

“通知”，表示 Aspect 在特定的 Join point 采取的操作。包括 “around”，“before” and “after” 等 Advice，大体上分为了三类：BeforeAdvice、MethodInterceptor、AfterAdvice

* **Advisor:**

> org.springframework.aop.Advisor

“通知者”，它持有 Advice，是 Spring AOP 的一个基础接口。它的子接口 PointcutAdvisor 是一个功能完善接口，它涵盖了绝大部分的 Advisor。

* **Advised:**

> org.springframework.aop.framework.Advised

AOP 代理工厂配置类接口。提供了操作和管理 Advice 和 Advisor 的能力。它的实现类 ProxyFactory 是 Spring AOP 主要用于创建 AOP 代理类的核心类。

****热插拔AOP执行核心逻辑****

![[图片](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/526d9d5d2bb94a4d82950ff363e48d3e~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=510&h=738&s=12978&e=webp&b=febf b)]

图片

****核心实现代码****

1、动态管理advice端点实现

```
...

@RestControllerEndpoint(id = "proxy")
@RequiredArgsConstructor
public class ProxyMetaDefinitionControllerEndPoint {

    private final ProxyMetaDefinitionRepository proxyMetaDefinitionRepository;

    @GetMapping("listMeta")
    public List<ProxyMetaDefinition> getProxyMetaDefinitions(){
        return proxyMetaDefinitionRepository.getProxyMetaDefinitions();
    }

    @GetMapping("/{id}")
    public ProxyMetaDefinition getProxyMetaDefinition(@PathVariable("id") String proxyMetaDefinitionId){
        return proxyMetaDefinitionRepository.getProxyMetaDefinition(proxyMetaDefinitionId);
    }

    @PostMapping("save")
    public String save(@RequestBody ProxyMetaDefinition definition){

        try {
            proxyMetaDefinitionRepository.save(definition);
            return "success";
        }
    }
}
```

```

        } catch (Exception e) {

        }
        return "fail";
    }

    @PostMapping("delete/{id}")
    public String delete(@PathVariable("id")String proxyMetaDefinitionId){
        try {
            proxyMetaDefinitionRepository.delete(proxyMetaDefinitionId);
            return "success";
        } catch (Exception e) {

        }
        return "fail";
    }
}
...

```

2、利用事件监听机制捕获安装或者卸载插件

```

...
@RequiredArgsConstructor
public class ProxyMetaDefinitionChangeListener {

    private final AopPluginFactory aopPluginFactory;

    @EventListener
    public void listener(ProxyMetaDefinitionChangeEvent proxyMetaDefini
tionChangeEvent){
        ProxyMetaInfo proxyMetaInfo = aopPluginFactory.getProxyMetaInfo
(proxyMetaDefinitionChangeEvent.getProxyMetaDefinition());
        switch (proxyMetaDefinitionChangeEvent.getOperateEventEnum()){
            case ADD:
                aopPluginFactory.installPlugin(proxyMetaInfo);
                break;
            case DEL:
                aopPluginFactory.uninstallPlugin(proxyMetaInfo.getId());
                break;
        }
    }
}
}

```

...

3、安装插件

...

```
public void installPlugin(ProxyMetaInfo proxyMetaInfo){
    if(StringUtils.isEmpty(proxyMetaInfo.getId())){
        proxyMetaInfo.setId(proxyMetaInfo.getProxyUrl() + SPIELT + proxyMetaInfo.getProxyClassName());
    }
    AopUtil.registerProxy(defaultListableBeanFactory,proxyMetaInfo);
}
```

...

4、安装插件核心实现

...

```
public static void registerProxy(DefaultListableBeanFactory beanFactory,
ProxyMetaInfo proxyMetaInfo){
    AspectJExpressionPointcutAdvisor advisor = getAspectJExpression
PointcutAdvisor(beanFactory, proxyMetaInfo);
    addOrDelAdvice(beanFactory,OperateEventEnum.ADD,advisor);
}
```

```
private static AspectJExpressionPointcutAdvisor getAspectJExpression
nPointcutAdvisor(DefaultListableBeanFactory beanFactory, ProxyMetaInf
o proxyMetaInfo) {
    BeanDefinitionBuilder builder = BeanDefinitionBuilder.genericBean
Definition();
    GenericBeanDefinition beanDefinition = (GenericBeanDefinition) bu
ilder.getBeanDefinition();
    beanDefinition.setBeanClass(AspectJExpressionPointcutAdvisor.cla
ss);
    AspectJExpressionPointcutAdvisor advisor = new AspectJExpressio
nPointcutAdvisor();
    advisor.setExpression(proxyMetaInfo.getPointcut());
    advisor.setAdvice(Objects.requireNonNull(getMethodInterceptor(pro
xyMetaInfo.getProxyUrl(), proxyMetaInfo.getProxyClassName())));
    beanDefinition.setInstanceSupplier((Supplier<AspectJExpressionPoi
ntcutAdvisor>) () -> advisor);
    beanFactory.registerBeanDefinition(PROXY_PLUGIN_PREFIX + prox
```

```
yMetaInfo.getId(),beanDefinition);
```

```
    return advisor;  
}
```

```
...
```

5、卸载插件

```
...
```

```
public void uninstallPlugin(String id){  
    String beanName = PROXY_PLUGIN_PREFIX + id;  
    if(defaultListableBeanFactory.containsBean(beanName)){  
        AopUtil.destoryProxy(defaultListableBeanFactory,id);  
    }else{  
        throw new NoSuchElementException("Plugin not found: " + id);  
    }  
}
```

```
...
```

6、卸载插件核心实现

```
...
```

```
public static void destoryProxy(DefaultListableBeanFactory beanFactory,  
String id){  
    String beanName = PROXY_PLUGIN_PREFIX + id;  
    if(beanFactory.containsBean(beanName)){  
        AspectJExpressionPointcutAdvisor advisor = beanFactory.getBe  
an(beanName,AspectJExpressionPointcutAdvisor.class);  
        addOrDelAdvice(beanFactory,OperateEventEnum.DEL,advisor);  
        beanFactory.destroyBean(beanFactory.getBean(beanName));  
    }  
}
```

```
...
```

7、操作advice实现

```
...
```

```
public static void addOrDelAdvice(DefaultListableBeanFactory beanFacto  
ry, OperateEventEnum operateEventEnum,AspectJExpressionPointcutAd
```

```

visor advisor){
    AspectJExpressionPointcut pointcut = (AspectJExpressionPointcut)
    advisor.getPointcut();
    for (String beanDefinitionName : beanFactory.getBeanDefinitionNa
mes()) {
        Object bean = beanFactory.getBean(beanDefinitionName);
        if(!(bean instanceof Advised)){
            if(operateEventEnum == OperateEventEnum.ADD){
                buildCandidateAdvised(beanFactory,advisor,bean,beanDefi
nitionName);
            }
            continue;
        }
        Advised advisedBean = (Advised) bean;
        boolean isFindMatchAdvised = findMatchAdvised(advisedBean.g
etClass(),pointcut);
        if(operateEventEnum == OperateEventEnum.DEL){
            if(isFindMatchAdvised){
                advisedBean.removeAdvice(advisor.getAdvice());
                log.info("#####
## Remove Advice --> 【{}】 For Bean --> 【{}】
SUCCESS !",advisor.getAdvice().getClass().getName(),bean.getClass().g
etName());
            }
        }else if(operateEventEnum == OperateEventEnum.ADD){
            if(isFindMatchAdvised){
                advisedBean.addAdvice(advisor.getAdvice());
                log.info("#####
## Add Advice --> 【{}】 For Bean --> 【{}】
SUCCESS !",advisor.getAdvice().getClass().getName(),bean.getClass().g
etName());
            }
        }
    }
}
}
}

```

...

****热插拔AOP演示示例****

1、创建一个service

...

@Service

@Slf4j

public class HelloService implements BeanNameAware, BeanFactoryAware {

private BeanFactory beanFactory;

private String beanName;

@SneakyThrows

public String sayHello(String message) {

Object bean = beanFactory.getBean(beanName);

log.info("===== {} is Advised : {}", bean, bean instanceof Advised);

TimeUnit.SECONDS.sleep(new Random().nextInt(3));

log.info("===== hello:{}", message);

return "hello:" + message;

}

@Override

public void setBeanFactory(BeanFactory beanFactory) throws BeansException {

this.beanFactory = beanFactory;

}

@Override

public void setBeanName(String name) {

this.beanName = name;

}

}

...

2、创建一个controller

...

@RestController

@RequestMapping("hello")

@RequiredArgsConstructor

public class HelloController {

private final HelloService helloService;

@GetMapping("/{message}")

///

22

场景一：未添加切面时 浏览器访问

: [http://localhost:8080/hello/zhangsan](http://cxyroad.com/
"http://localhost:8080/hello/zhangsan") 观察控制台

![图片](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/15453f384cf14043abf702f291192d0e~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=418&h=42&s=2194&e=webp&b=212a32)

图片

****场景二：通过postman动态操作代理****

1、新增代理

![图片](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/7681ba628f854f5588e9a458aef21cf3~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=469&h=280&s=12040&e=webp&b=fcfcfc)

图片

观察控制台

```
...  
##### BuildCandidateAdvised --> 【com.github.lybgeek.aop.test.hello.service.HelloService】  
With Advice --> 【  
com.github.lybgeek.interceptor.LogMethodInterceptor】 SUCCESS !  
...
```

此时浏览器访问

: [http://localhost:8080/hello/zhangsan](http://cxyroad.com/
"http://localhost:8080/hello/zhangsan")

再次观察控制台

![图片](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/a0584db383274e5abee1516ceb4848a6~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1080&h=91&s=12356&e=webp&b=1f2931)

图片

出现了切面日志信息，说明代理生效

2、删除代理

![图片](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/0280018ecf0d4cb6a50d08602711362e~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=397&h=293&s=5360&e=webp&b=fdfdfd)

图片

观察控制台

```
...  
##### Remove Advice --  
> 【com.github.lybgeek.interceptor.LogMethodInterceptor】 For Bean --  
> 【  
com.github.lybgeek.aop.test.hello.service.HelloService$$EnhancerBySpringCGLIB$$7bc75aa3】 SUCCESS !  
...
```

此时浏览器访问

: [http://localhost:8080/hello/zhangsan](http://cxyroad.com/"http://localhost:8080/hello/zhangsan")

再次观察控制台

![图片](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/1805cb3891f6456f98fb7699d377e8f8~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=423&h=36&s=2012&e=webp&b=202a32)

)

图片

此时没有出现切面日志信息，说明代理删除成功

****总结****

本文实现热插拔AOP就在于对advice、advised、advisor、pointcut概念的理解，这是实现热插拔AOP的前提。

最后说一句(求!别白嫖!)

如果这篇文章对您有所帮助，或者有所启发的话，求一键三连：点赞、转发、在看。

公众号：****woniuxgg****，在公众号中回复：笔记 就可以获得蜗牛为你精心准备的java实战语雀笔记，回复面试、开发手册、有超赞的粉丝福利！

原文链接: <https://juejin.cn/post/7385784332445155339>