xmlns:aop="http://www.springframework.org/schema/aop"

http://www.springframework.org/schema/aop/spring-aop.xsd

启用

@AspectJ-style

@Aspect

@Around(*PointCutStr*)

public Object aroundFunc(JoinPoint joinPoint) throws Throwable {

}

import org.aspectj.lang.JoinPoint;  
import org.aspectj.lang.ProceedingJoinPoint;  
import org.aspectj.lang.annotation.Around;  
import org.aspectj.lang.annotation.Aspect;

<aop:aspectj-autoproxy />

proxy-target-class="true"

Are class-based (CGLIB) proxies to be created? By default, standard Java interface-based proxies are created.

standard:class com.sun.proxy.$Proxy13

CGLib:class com.jiangli.common.core.HelloImpl$$EnhancerBySpringCGLIB$$ce5f64a2

切面逻辑

*<!-- 记录方法运行时长日志 -->*<bean id="runningTimerAspect" class="com.zhihuishu.treenity.log.RunningTimerAspect" />

切面配置

*<!-- 日志切面配置 -->*<aop:config>  
 <aop:pointcut id="allPublicMethodPointcut"   
 expression="execution(public \* com.zhihuishu.treenity.service..\*.\*(..)) or execution(public \* com.zhihuishu.treenity.mapper..\*.\*(..))" />  
 <aop:aspect id="runningTimerAspectAspect" ref="runningTimerAspect">  
 <aop:around method="around" pointcut-ref="allPublicMethodPointcut" />  
 </aop:aspect>  
</aop:config>