Atitit aop 基于spring 注解 实现 操作日志 attilax总结

[1.1. 调用方 设置注解 @LogAnno spring模式 1](#_Toc16733)

[1.2. 非spring模式注解 1](#_Toc9827)

[1.3. 设置切面 2](#_Toc13475)

[1.4. Spring xml配置 4](#_Toc23385)

[1.5. 调用测试 spring模式 5](#_Toc5767)

[1.6. 非spring模式anno解析实现 6](#_Toc11260)

[1.7. 非spring模式调用测试 7](#_Toc22480)

[2. 参考资料 8](#_Toc3111)

## 调用方 设置注解 @LogAnno spring模式

Op属性指明操作动作。。。 setTmpParam注入当前用户名

|  |
| --- |
| @Async // 指明异步操作  @LogAnno(value = "操作日志记录:{u}在{do}了", op = "someop")  **public** **int** someOp(**final** **int** i) {  Global.*setTmpParam*("user", "ati9");  **return** i + 3;  } |

## 非spring模式注解

|  |
| --- |
| @secury  // @Value(value = "--\*\*操作日志记录:{u}在{do}了")  @Async // 指明异步操作  @LogAnno(value = " ", op = "someop")  **public** **void** someOp(**final** **int** i, **final** String para2) {  System.***out***.println(i);  Map **m** = **new** HashMap<>();  m.put("user", "ati");  m.put("params", **new** HashMap() {  {  **this**.put("p1\_int", String.*valueOf*(i));  **this**.put("p2\_str", para2);  }  });  **new** AnnoProcessor().parse(m);  } |

## 设置切面

/cloudcommons/src/main/java/com/cnhis/cloudhealth/commons/aoplog/MyLogAspect.java

|  |
| --- |
| **package** com.cnhis.cloudhealth.commons.aoplog;  **import** org.aspectj.lang.JoinPoint;  **import** org.aspectj.lang.ProceedingJoinPoint;  **import** org.aspectj.lang.annotation.\*;  **import** org.aspectj.lang.reflect.MethodSignature;  **import** org.springframework.stereotype.Component;  **import** com.alibaba.fastjson.JSON;  **import** com.cnhis.cloudhealth.commons.aoplog.anno.AnnoProcessor;  **import** com.cnhis.cloudhealth.commons.aoplog.anno.LogAnno;  **import** com.cnhis.cloudhealth.commons.aoplog.anno.annoTest;  **import** com.cnhis.cloudhealth.commons.aoplog.util.Global;  **import** com.cnhis.cloudhealth.commons.aoplog.util.PathUtil;  **import** com.cnhis.cloudhealth.commons.aoplog.util.SpringUtil;  **import** java.lang.reflect.Method;  **import** java.util.HashMap;  **import** java.util.Map;  /\*\*  \* **@author** sam  \* **@since** 2017/7/13  \*/  @Aspect // AOP 切面  @Component  **public** **class** MyLogAspect {  **public** **static** **void** main(String[] args) {  // attilax 老哇的爪子 上午11:54:26 2014-5-14  // SpringUtil.cfgFileDir=PathUtil.classPath\_hisCommLib()+"/";  SpringUtil.*locations* = **new** String[] { PathUtil.*classPath\_hisCommLib*() + "/" + "IocSrpingCfg.xml" };  annoTest **bean** = (annoTest) SpringUtil.*getBean*("annoTest");  bean.someOp(56);  System.***out***.println(bean);  System.***out***.println("--");  // branchManagerService  }  // 切入点  @Pointcut(value = "@annotation( com.cnhis.cloudhealth.commons.aoplog.anno.LogAnno)")  **private** **void** pointcut() {  }  @SuppressWarnings("all")  @AfterReturning(value = "pointcut() && @annotation(myLog)", returning = "result")  **public** Object afterReturning(JoinPoint joinPoint, LogAnno myLog, Object result) {  // HttpServletRequest request = ((ServletRequestAttributes)  // RequestContextHolder.getRequestAttributes()).getRequest();  // HttpSession session = request.getSession();  System.***out***.println("++++执行了afterReturning方法++++");  System.***out***.println("执行结果：" + result);  // 类名  String **clazzName** = joinPoint.getTarget().getClass().getSimpleName();  // 得到方法名  String **methodName** = joinPoint.getSignature().getName();  MethodSignature **MethodSignature1** = (MethodSignature) joinPoint.getSignature();  // 入参key  **final** String[] **parameterNames** = MethodSignature1.getParameterNames();  // 入参value  **final** Object[] **arguments** = joinPoint.getArgs();  Method **method** = MethodSignature1.getMethod();  // 方法的注解对象  LogAnno **logParam** = method.getAnnotation(LogAnno.**class**);  Map **m** = **new** HashMap<>();  m.put("user", Global.*tempParamMap*.get().get("user"));  m.put("op", logParam.op());  m.put("class", clazzName);  m.put("method", methodName);  m.put("params", **new** HashMap() {  {  // 入参字符串  **for** (**int** **i** = 0; i < parameterNames.length; i++) {  String **paramname\_cur** = parameterNames[i];  Object **para\_val\_object** = arguments[i];  **this**.put(paramname\_cur, para\_val\_object);  }  }  });  m.put("return", result);  **new** OperLogUtil().log4postgre( m);  // String json="{param:@p@}".replace("@p@", String.valueOf( i));  // new AnnoProcessor().parse(m);  **return** result;  } |

## Spring xml配置

/cloudcommons/src/main/java/IocSrpingCfg.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:aop="http://www.springframework.org/schema/aop"  xmlns:tx="http://www.springframework.org/schema/tx" xmlns:context="http://www.springframework.org/schema/context"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans-3.1.xsd  http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx.xsd  http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop.xsd http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd"  default-lazy-init="true">  <!-- 启动对@AspectJ注解的支持 -->  <aop:aspectj-autoproxy proxy-target-class="true" />  <!-- 自动扫描包路径 -->  <context:component-scan base-package="com.iflytek.zhbs.common.aoplog" />  <context:component-scan base-package="com.iflytek.zhbs.service" />  <context:component-scan base-package="com.cnhis.cloudhealth.commons.aoplog.\*" />  <context:component-scan base-package="com.cnhis.cloudhealth.commons.aoplog.\*\*" />  <context:component-scan base-package="com.cnhis.cloudhealth.commons.aoplog.anno" />  <context:component-scan base-package="com.cnhis.cloudhealth.commons.aoplog" />  </beans> |

## 调用测试 spring模式

|  |
| --- |
| **public** **class** MyLogAspect {  **public** **static** **void** main(String[] args) {  // attilax 老哇的爪子 上午11:54:26 2014-5-14  // SpringUtil.cfgFileDir=PathUtil.classPath\_hisCommLib()+"/";  SpringUtil.*locations* = **new** String[] { PathUtil.*classPath\_hisCommLib*() + "/" + "IocSrpingCfg.xml" };  annoTest **bean** = (annoTest) SpringUtil.*getBean*("annoTest");  bean.someOp(56);  System.***out***.println(bean);  System.***out***.println("--");  // branchManagerService  } |

## 非spring模式anno解析实现

|  |
| --- |
| **public** **void** parse(Map m2) {  **try** {  Thread **currentThread** = Thread.*currentThread*();    StackTraceElement[] **stackTrace** = currentThread.getStackTrace();  StackTraceElement **StackTraceElement1\_invoker** = stackTrace[2];  // StackTraceElement1\_invoker.  Class **cls** = Class.*forName*(StackTraceElement1\_invoker.getClassName());  String **methodName** = StackTraceElement1\_invoker.getMethodName();  // Method m=cls.getm  Method[] **methods** = cls.getDeclaredMethods();  // System.err.println( JSON.toJSONString(stackTrace,true));  **for** (Method **method** : methods) {      **if** (method.isAnnotationPresent(Value.**class**) && method.getName().equals(methodName)) {  Value **LogAnno1** = method.getAnnotation(Value.**class**);  String **LogAnno1\_val\_tmp** = LogAnno1.value();  String **u**=(String) m2.get("user");  LogAnno1\_val\_tmp=LogAnno1\_val\_tmp.replaceAll("\\{u\\}", u);  //LogAnno1\_val\_tmp=LogAnno1\_val\_tmp.replaceAll("\\{do\\}", dox);  String **clazzName** = cls.getName();  m2.put("class", clazzName);  m2.put("method", methodName);  System.***out***.println(  "clazzName: " + clazzName + ", methodName: " + methodName + ", do: " + LogAnno1\_val\_tmp);  LogAnno1\_val\_tmp=JSON.*toJSONString*(m2);  **new** OperLogUtil().log4postgre( m2);  }  }  } **catch** (Exception **e**) {  // **TODO** Auto-generated catch block  e.printStackTrace();  }  // Method m= tAnno.class.getMethod(stackTrace[1].getMethodName(),  // String[].class);  // sql a2 =m.getAnnotation(sql.class);  // System.out.println(a2.value());    } |

## 非spring模式调用测试

|  |
| --- |
| **package** com.cnhis.cloudhealth.commons.aoplog.anno;  **import** java.lang.annotation.Retention;  **import** java.lang.annotation.RetentionPolicy;  **import** java.util.HashMap;  **import** java.util.Map;  **import** org.springframework.beans.factory.annotation.Value;  **import** org.springframework.scheduling.annotation.Async;  **import** org.springframework.scheduling.annotation.EnableAsync;  **import** org.springframework.stereotype.Component;  **import** com.cnhis.cloudhealth.commons.aoplog.util.Global;  @gbk编码  @LogAnno(op = "", value = "")  @Component  **public** **class** annoTest {  **public** **static** **void** main(String[] args) {  **new** annoTest().someOp(22, "para2val");  System.***out***.println("--");  } |

# 参考资料

Spring注解实现日志记录 - CSDN博客.html