

## 一、功能说明

该工具包的作用是实时记录后台用户对数据的操作记录, 追踪恶意操作, 追查不良行为。具有轻量级、高扩展性等特点。

### 背景

- 当前项目大都具备后台管理模块, 且配备多用户多角色模式
- 后台用户的失误/恶意操作难以追踪, 因此建设该通用工具提供操作记录的解决方案。

## 二、工具使用

- 添加依赖: pom.xml

```
<dependency>
  <groupId>cn.gmlee</groupId>
  <artifactId>tools-datalog</artifactId>
  <version>3.1.1</version>
</dependency>
```

- 启用工具

在启动类上或其他配置类上扫描包: cn.gmlee.tools

```
@SpringBootApplication
@ComponentScan("cn.gmlee.tools")
public class XxxApp {
    public static void main(String[] args) {
        SpringApplication.run(XxxApp.class, args);
    }
}
```

- 添加注解: @ApiPrint

```
// 非必需
@ApiPrint("接口说明")
@RequestMapping("...")
public JsonResult come(Object... os) {
    return JsonResult.OK;
}
```

- 初始化表: datalog.sql

可以自主扩展列、更改表名、忽略实体等操作, 高阶用法请联系@Auther

```
SET FOREIGN_KEY_CHECKS=0;

-----
-- Table structure for datalog
-----

DROP TABLE IF EXISTS `datalog`;
```

```
CREATE TABLE `datalog` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '主键',
  `start_ms` bigint(20) DEFAULT NULL COMMENT '开始记录时间',
  `total_ms` bigint(20) DEFAULT NULL COMMENT '记录完成耗时',
  `original_sql` text COMMENT '更新原始sql',
  `where_sql` text COMMENT '条件语句',
  `old_data_json` text COMMENT '旧数据',
  `new_data_json` text COMMENT '新数据',
  `datalog_select_sql` text COMMENT '查询旧数据所用语句',
  `data_table` varchar(64) DEFAULT NULL COMMENT '更新数据表名',
  `data_log` text COMMENT '更新日志',
  `api` varchar(32) DEFAULT NULL COMMENT '请求地址',
  `site` varchar(255) DEFAULT NULL COMMENT '请求位置',
  `params` text COMMENT '请求参数',
  `request_ip` varchar(16) DEFAULT NULL COMMENT '请求IP',
  `request_url` varchar(255) DEFAULT NULL COMMENT '请求url',
  `request_time` datetime DEFAULT NULL COMMENT '请求时间',
  `user_id` bigint(20) DEFAULT NULL COMMENT '用户编号',
  `username` varchar(64) DEFAULT '游客' COMMENT '用户名称',
  `create_at` datetime DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;
```

### 三、工具原理

#### 逻辑

- 该工具通过AOP实现接口信息记录
- 通过mybatis拦截器完成数据更新监控
- 如果是update操作则收集当前线程上的新/旧数据再在提交时处理数据

注意: 收集旧数据时默认采集的事务隔离级别是: READ\_COMMITTED

因此: 数据源需要关闭自动提交(加@Transactional注解也能达到效果)

否则: **采集到的数据均为新数据** (因为业务程序自动提交了事务)

- 基于GM-TOOLS完成数据库操作

#### 代码

1. 通过DatalogAspect切面完成接口信息的记录
2. 通过DatalogInterceptor拦截器实现数据监控
3. 在cn.gmlee.tools.datalog.interceptor.DatalogInterceptor.datalog中可见该工具对select操作无感
4. 在同类recordUpdateDatalog方法中可见使用@Ignore可以忽略指定的实体

## 数据

- 接口数据与数据日志均保存在内存中
- 接口数据在接口的最终通知中始终被清理
- 数据日志在每次请求classForDatalog()时被清理

## 四、使用示例

- 日志样本

```
(3047):  
{  
  逻辑删除:[false]->[false]  
  车辆类型{MINIBUS:1:小面,MICROBUS:2:中面,MINI_TRUCK:3:小货,MEDIUM_TRUCK:4:中  
货}: [null]->[2]  
  修改时间:[2021-06-18 16:38:51.0]->[2021-7-10 14:20:37]  
  修改人:[1885]->[1889]  
}
```

- 日志说明

```
(主键):{  
  列名: 旧数据 -> 新数据  
}
```