



链滴

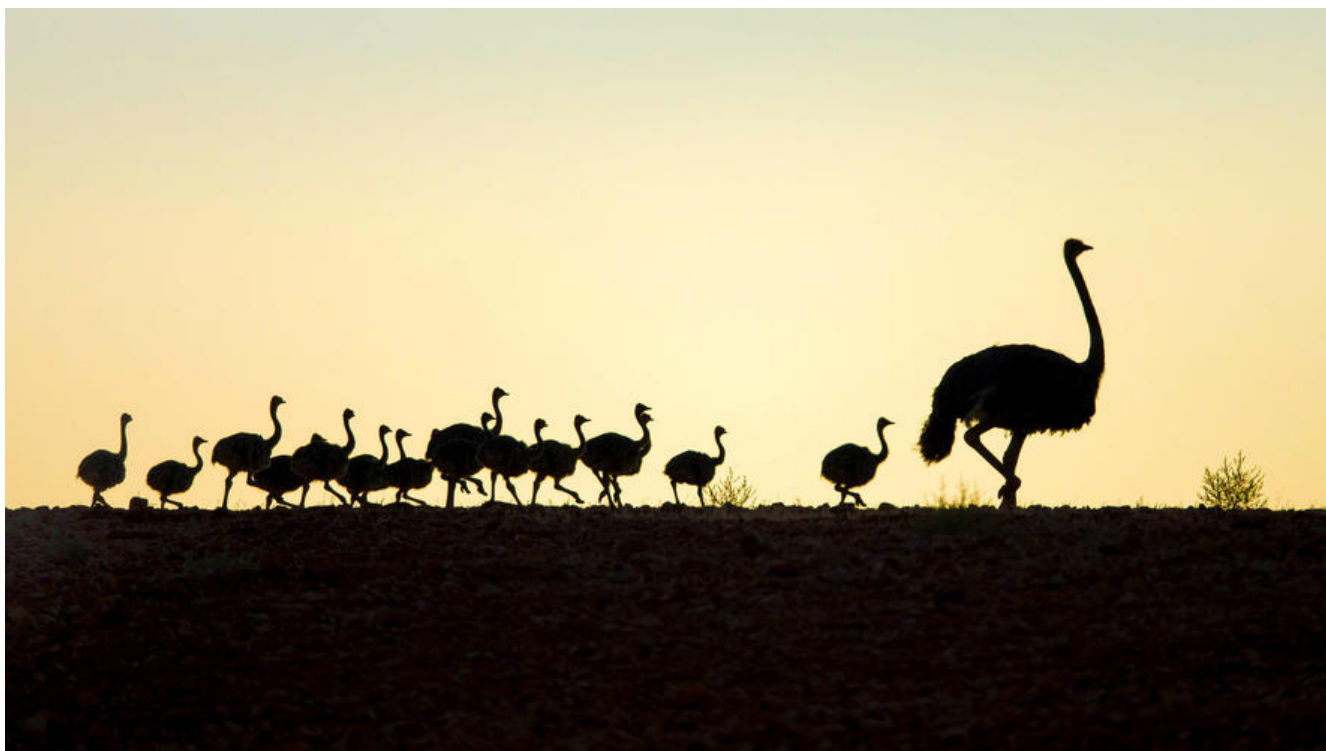
Springboot+mybatis 最简单的增删改查写法

作者: [xiaodaojava](#)

原文链接: <https://ld246.com/article/1597023508598>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



增删改查估计是每个程序员都痛恨的事情,其实就是复制粘贴,于是乎,小刀就折腾了一个工具类,可以帮大家最大程度上(至少我觉得很简化)去少些增删改查代码

引入依赖

除了常用的springboot相关依赖,要加上小刀打的工具包,整体依赖如下

```
implementation 'org.springframework.boot:spring-boot-starter-web'  
implementation 'org.springframework.boot:spring-boot-starter-data-redis'  
implementation 'org.mybatis.spring.boot:mybatis-spring-boot-starter:2.1.3'
```

```
// 小刀打的工具包依赖(已发布到中央仓库)  
compile 'red.lixiang.tools:sunshine-starter:1.0.1'  
// 这里方便测试,引了h2,实际中可以换成mysql  
runtimeOnly 'com.h2database:h2'
```

准备测试表

这里为了代码运行方便,我们直接引入了h2数据库.表结构及测试数据如下:

```
drop table users if exists;
```

```
create table users (  
    id int,  
    name varchar(20)  
);
```

```
insert into users (id, name) values((select count(*) + 1 from users), 'XiaoDao');  
insert into users (id, name) values((select count(*) + 1 from users), '公众号:java技术大本营');  
insert into users (id, name) values((select count(*) + 1 from users), '微信:best396975802');
```

编写实体类

这里的实体类有两个,一个是和数据库对应的实体类,一个是查询用的QC,QC中封装了一些和查询相关的通用字段.

数据库对应字段:

Users:

```
public class Users {
    // 这个字段标识了是和数据库对应的字段
    @SqlField
    private Long id;
    @SqlField
    private String name;

    // 这里省略了get/set方法
}
```

UsersQC:

```
// 注意这里要继承一下BaseQC
public class UsersQC extends BaseQC {

    private Long id;
    // 这里的QC注解,按需使用,如这里,标识name是一个可以模糊查询的字段
    @QC(likeQuery = true)
    private String name;

}
```

编写mybatis的mapper文件

```
// 需要继承BaseMapper,里面封装了基本的增删改查的写法
@Mapper
@Repository
public interface UsersMapper extends BaseMapper<Users> {

    // 需要重写这个方法,去完成一些缓存的初始化
    @Override
    default Class<?> getMapperClass() {
        MybatisToolCache.cacheDomain("users",Users.class,UsersQC.class);

        return UsersMapper.class;
    }
}
```

编写spring的配置文件

这里面没有什么单独的配置,都是h2数据库的配置,在实际使用中,可以替换成mysql的数据库

#配置数据库连接地址

```
spring.datasource.platform=h2
#配置数据库驱动
spring.datasource.driver-class-name=org.h2.Driver
#配置数据库用户名
spring.datasource.username=sa
spring.datasource.schema=classpath:test_scheme.sql
spring.datasource.data=classpath:test_data.sql
```

启动项目

到目前为止,只需要我们写这些代码, Controller小刀直接封装在框架中了,启动项目之后,可以访问 <http://127.0.0.1/simple/users/query> 看到对应的数据库数据.

同样封装的请求还有获取单个的

<http://127.0.0.1/simple/users/get?id=1>

新增/保存的

<http://127.0.0.1/simple/user/save> (post请求)

删除的

<http://127.0.0.1/simple/remove?id=1>

源码地址

上述文章的代码地址:

<https://github.com/xiaodaojava/sunshine-sample>

工具类的源码地址:

<https://github.com/xiaodaojava/sunshine>

总结

在使用过程中,有什么疑问,欢迎随时私信小刀一起讨论交流~~