# Lucky使用文档

## 环境

1. 导入jar包

Mysql的驱动包和Lucky的jar包



1. 创建配置文件

名称：lucky.properties（固定名称）  
位置：src文件夹下



配置内容：

1.**数据库部分：**

jdbc.driver=com.mysql.jdbc.Driver

jdbc.url=jdbc:mysql:/ip:port://数据库名

jdbc.username=用户名

jdbc.password=密码

**2.控制台信息是否打印的配置：**

\*\*\*可缺省，缺省则默认不打印信息\*\*\*

Log=true

**3.Lucky缓存机制的配置：**

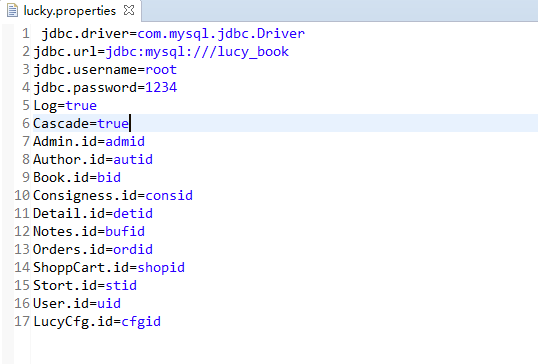
\*\*\*可缺省，缺省则默认不使用缓存机制\*\*\*

Cache=true

**4.ID操作配置（ID查询和删除）：**

\*\*\*可缺省，缺省则无法使用配置方式的ID操作\*\*\*

类名.id=主键对应的属性



## 使用SqlControl类操作数据库

### 不使用Lucky的建表机制

#### 1.实体类的创建规范

**1. 实体类的类名必须与表名一致**

**2. 实体类的属性名必须与表的字段名一致**

**3. 实体类的属性的类型必须与表字段的类型一致**

**4. 表的主键必须为int类型**

**5. 实体类的属性类型必须为java基本类型的包装类型**

**6. 每个实体类都必须由系统生成toString()方法**

#### 2.使用SqlControl类的无参构造获得对象

**SqlControl sql=new SqlControl（）;**

#### 3.执行增加操作

1. **预编译sql方式**



**例：**

**sql.save(“INSERT INTO Book(bname,price) VALUES(?,?)”,”《朝花夕拾》”,23.5);**

**2． 对象方式**



**例：**

**Book book=new Book();**

**book.setBname(“《朝花夕拾》”);**

**book.setPrice(23.5);**

**sql.save(book);**

#### 4.执行修改操作

1. **预编译sql方式**



**例：**

**sql.update(“UPDATE Book SET price=? WHERE bid=?”,44.6,2);**

**2． 对象方式**



**例：**

**Book book=new Book();**

**book.setBid(2);**

**book.setPrice(44.6);**

**sql.update(book);**

#### 5.执行删除操作

**1． 预编译sql方式**



**例：**

**sql.delete(“DELETE FROM Book WHERE stort=? and price=?”,”网络小说”,43);**

**2． 对象方式**



**例：**

**Book book=new Book();**

**book.setStort(“网络小说”);**

**book.setPrice(43);**

**sql.delete(book);**

**3． ID方式（配置）**

\*\*\*此操作需要在配置文件中设置ID配置\*\*\*



**例：**

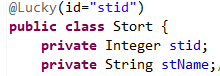
**sql.delete(“Book”，2);**

**4． ID方式(注解)**

\*\*\*此操作需要使用注解@Lucky(id=”主键对应的属性”)\*\*\*



**例：**



**sql.delete(Srort.class，2);**

#### 6.执行查询操作

**1． 预编译sql方式**



**例：**

**List<Book> list=(List<Book>)sql.update(“SELECT FROM Book WHERE stort=? And price=?”,”小说”，44);**

**2． 对象方式**



**例：**

**Book book=new Book();**

**book.setStort(“小说”)；**

**book.setPrice(44)**

**List<Book> list =(List<Book>) sql.getList(book);**

**3． ID方式（配置）**



**例：**

**Stort stort=(Stort) sql.getOne(Stort.class,5);**

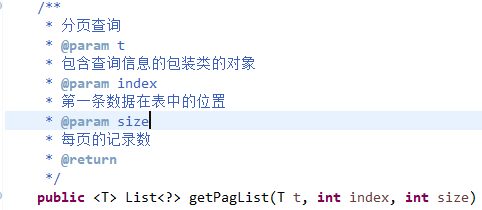
**4． ID方式(注解)**



**例：**

**Stort stort=(Stort) sql.getone(Stort.class,5);**

**5． 对象方式分页查询**



**例：**

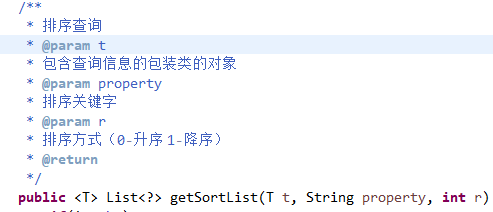
**Student student=new Student();**

**student.setSsex(“男”);**

**List<Student> list=sql.getPagList(student,0,5);**

**SELECT FROM Student WHERTE ssex=”男” LIMIT 0,9**

**6． 对象方式排序查询**



**例：**

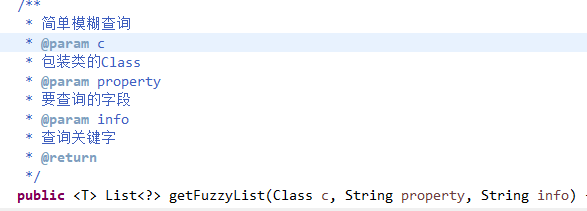
**Student student=new Student();**

**student.setSsex(“男”);**

**List<Student> list=sql.getSortList(student,”sname”,0);**

**SELECT FROM Student WHERTE ssex=”男” ORDER BY sname ASC**

**7． 对象方式简单模糊查询**



**例：**

**List<Student> list=sql.getFuzzyList(Student.class,”sname”,”张”)；**

**SELECT FROM Student WHERTE sname LIKE %张%**

### (二) 使用Lucky的建表机制

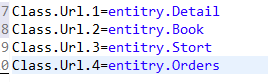
#### 建表的相关配置

1. 配置

使用Lucky的建表机制必须在配置文件中做实体类的映射配置

**Class.Url.标号=“实体类的全路径**”

**注：标号必须为连续的整数**



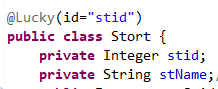
B．注解@Lucky

说明：建表注解，配置主键ID信息，和外键key信息

使用：

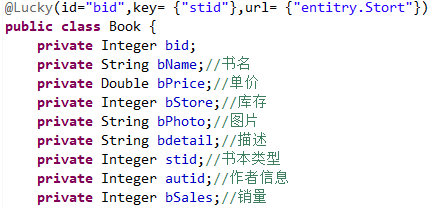
1. 没有外键

@Lucky(id=”ID属性名”)

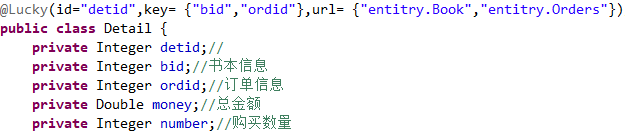


1. 单个外键

@Lucky(id=” ID属性名”,key={“外键属性名”},url={“对应父表的类的全路径”})



1. 多个外键
2. @Lucky(id=” ID属性名”,key={“外键属性名1”,…, “外键属性名n”},url={“对应父表的类1的全路径” ,…,” 对应父表的类1的全路径”})



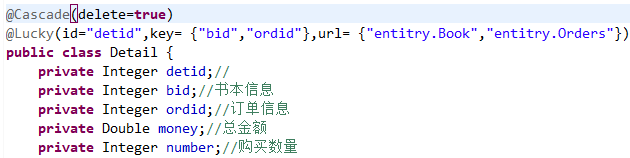
C．注解@Cascade

说明：为表设置级联外键

用法：

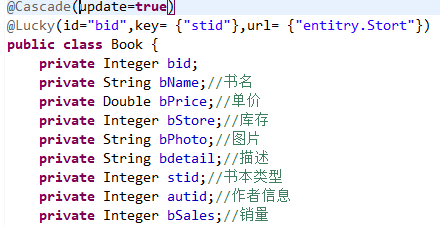
1. 级联删除

@Cascade(delete=true)



1. 级联更新

@Cascade(update=true)

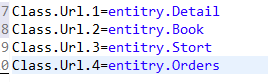


#### 使用SqlControl的有参构造建表



**SqlControl sql=new SqlControl（配置文件中第一个标号，最后一个标号）;**

例：



**SqlControl sql=new SqlControl（1，4）；**

### (三) Lucky的查询缓存

说明：使用Lucky的缓存机制必须在配置文件中将缓存机制设置为开启

当用户第一次执行查询操作时，lucky会向mysql数据库发送sql语句，得到查询结果后会将结果存入缓存区，当用户再次执行查询操作时lucky先会去缓存区查找，如过有结果则直接返回，没有结果则向mysql发送sql语句。当用户执行非查询操作时，lucky会自动提取操作的关键字，并会根据关键字清除缓存中可能会受此操作影响的那些结果。

### (四) Lucky的事务处理

Transaction类

Transaction tx=sql.openTransaction();过的事务对象并开启事务

tx.commit(); 提交事务

tx.rollback()；事务回滚