

目录

- 1. 实现联表查询的一个小技巧
 - 1.1 左联表查询SQL语句
 - 1.2 Druid数据库连接池中url地址的写法
- 2. 商品详情页对RestFul风格的使用
- 3. 秒杀功能实现的逻辑
 - 3.1 减少库存的sql语句
 - 3.2 创建订单的sql语句
 - 3.3 @SelectKey()注解
 - 3.3.1 获取主键值的注意事项

1. 实现联表查询的一个小技巧

商品表

秒杀商品表

商品表和秒杀商品表是两个互相独立的表，其中的关联为 `goods_id`，但是我要返回的对象，既想要商品表中的字段，又想要秒杀商品表中的字段，用下面这个方法，有点儿亮眼

```
1  @Data
2  public class GoodsVo extends Goods {
3      private Double miaoshaPrice;
4      private Integer stockCount;
5      private Date startDate;
6      private Date endDate;
7  }
```

创建一个GoodsVo类，继承Goods类，再将 秒杀商品表中特有的字段，添加进去即可

1.1 左联表查询SQL语句

- 查询所有的商品

```
1 | @Select("select g.*,mg.stock_count,mg.miaosha_price,mg.start_date,mg.end_date from mi  
2 | public List<GoodsVo> listGoodsVo();
```

- 根据id获取商品

```
1 | @Select("select g.*,mg.stock_count,mg.miaosha_price,mg.start_date,mg.end_date from mi  
2 | public GoodsVo getGoodsVoByGoodsId(@Param("goodId")long goodId);
```

1.2 Druid数据库连接池中url地址的写法

```
1 | jdbc:mysql://xxx.xx.xxx.xxx:3306/miaosha?useUnicode=true&characterEncoding=utf-8&allowMul
```

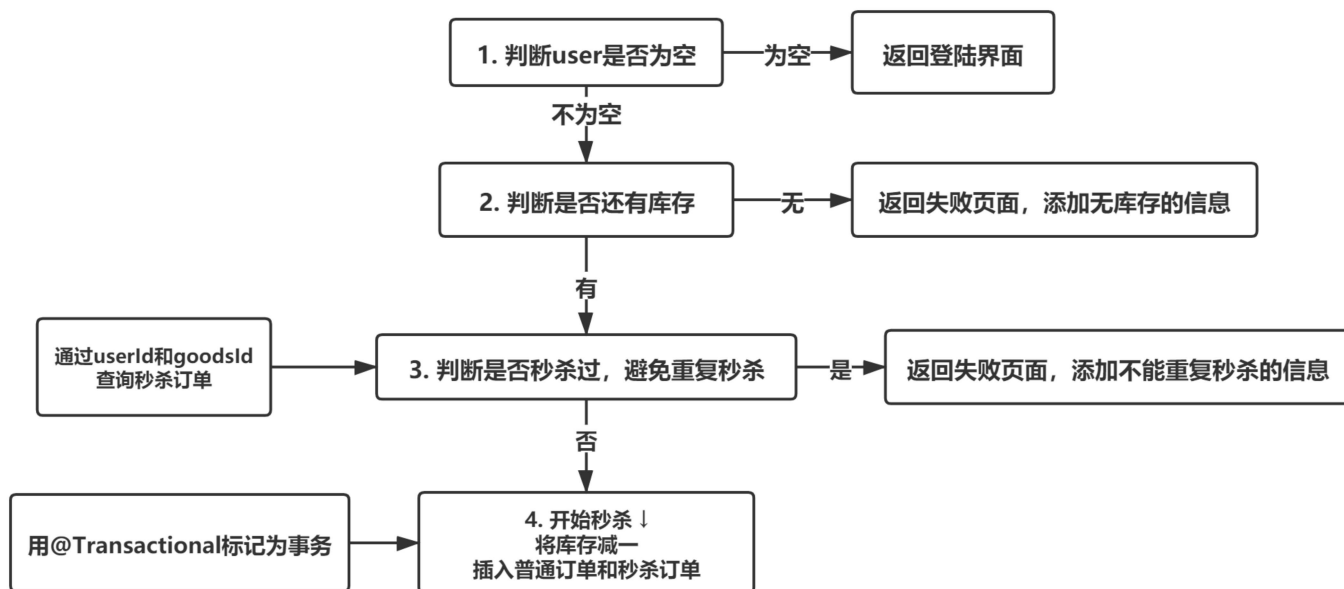
- `serverTimezone=GMT%2B8` :时区为GMT%2B8

2. 商品详情页对RestFul风格的使用

```
1 | @RequestMapping("to_detail/{goodsId}")  
2 | public String toDetail(Model model, MiaoShaUser user, @PathVariable("goodsId") long g  
3 | ...  
4 | }
```

- `@RequestMapping` 指定的映射URL，其中有用{}括起来的参数，在方法的形参处，用 `@PathVariable` 注解对其进行获取

3. 秒杀功能实现的逻辑



3.1 减少库存的sql语句

```

1 | @Update("update miaosha_goods set stock_count = stock_count - 1 where goods_id = #{go
2 | void reduceStock(MiaoshaGoods miaoshaGoods);

```

3.2 创建订单的sql语句

```

1 | @Insert("insert into order_info (user_id,goods_id,goods_name,goods_count,goods_price,
2 |       "values("#{userId},#{goodsId},#{goodsName},#{goodsCount},#{goodsPrice},#{order
3 | @SelectKey(statement = "select last_insert_id()",keyProperty = "id",resultType = long
4 | long insert(OrderInfo orderInfo);

```

3.3 @SelectKey()注解

- 需要前置注解：@Insert 或 @InsertProvider 或 @Update 或 @UpdateProvider，否则无效。
- statement：填入将会被执行的 SQL 字符串
- keyProperty属性：填入将会被更新的参数对象的属性
- before属性：填入 true 或 false 以指明 SQL 语句应被在插入语句的之前还是之后执行
- resultType属性：填入 keyProperty 的 Java 类型

3.3.1 获取主键值的注意事项

可能我们在执行完插入方法后，想如下这样通过获取返回值的方法来获取主键id值

```

1 | long orderId = orderDao.insert(orderInfo)

```

然而并不是这样， 因为执行插入sql语句返回值只有两种情况，一种是插入成功返回1，另一种是插入失败，返回0，所以若这里出现一直获取到1值的问题，用如下方法解决

```
1 | orderInfo.getId();
```

因为，@SelectKey()会将值直接映射到实体类的属性上进行修改，要想获取主键值，只能这样获取，不能通过返回值获取