

基于 MyBatis 的关联实现方案

一、实验目的

- 1、掌握 MyBatis 的 ResultSet 的使用方法
- 2、掌握用 SpringBoot 应用 Dao 层的实现方法
- 3、对在 MyBatis 中用不同方式实现数据表关联的效率比较

二、实验环境

- 1、服务器 A: Ubuntu 18.04 服务器 2 核 4G 内存虚拟机一台，图形界面，安装 JDK 11，Maven、git
- 2、服务器 B: Ubuntu 18.04 服务器 2 核 2G 内存虚拟机一台，命令行界面，安装 JDK 11，Maven、git，JMeter 5.4.1
- 3、服务器 C: Ubuntu 18.04 服务器 2 核 2G 内存虚拟机一台，命令行界面，安装 JDK 11，Maven、git，MySQL 8.0

三、实验内容及要求

- 在基于 MyBatis 的 SpringBoot 应用中，表的关联可以用两个方案完成
- 1. 在 MyBatis 中用 SQL 完成表的关联，返回关联的对象模型
 - 2. 在 MyBatis 中单独查询一个对象，在 Dao 层完成对象的关联要求
- 要求设计实验方案，分别用以上两个方案实现以下 RESTful API，比较两者的速度差异。

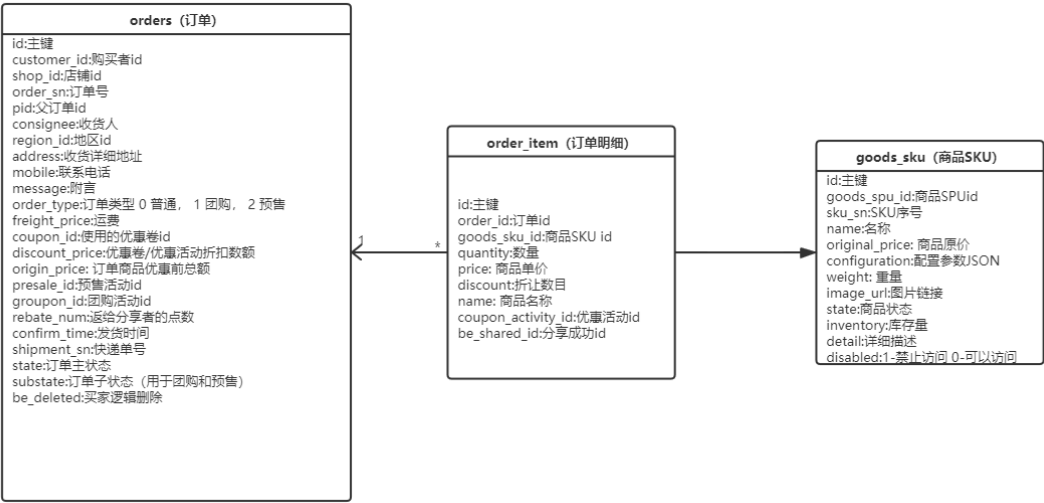
API	API 描述链接
查询订单完整信息	<div>GET /orders/{id}</div> <div>返回值:</div> <div><pre>{ "errno": 0, "errmsg": "成功", "data": { "id": 0, "orderSn": "string", "pid": 0, "orderType": 0, "state": 0, "subState": 0, "gmtCreate": "string", "gmtModified": "string", "confirmTime": "string", "originPrice": 0, "discountPrice": 0, "freightPrice": 0, "rebateNum": 0, "message": "string",</pre></div>

```

    "regionId": 0,
    "address": "string",
    "mobile": "string",
    "consignee": "string",
    "couponId": 0,
    "grouponId": 0,
    "presaleId": 0,
    "shipmentSn": "string",
    "orderItems": [
      {
        "skuId": 0,
        "orderId": 0,
        "name": "string",
        "quantity": 0,
        "price": 0,
        "discount": 0,
        "couponActId": 0,
        "beSharedId": 0
      }
    ]
  }
}
```

其中数据库的 ER 图如下图所示：

实验二，三ER图：



四、实验报告及提交要求

提交实验程序的 git 地址、实验原始数据（JMeter 的 jtl 文件）和实验报告。提交要求是实验程序和实验原始数据必须得到老师的肯定，实验报告必须能清晰说明实验设计、实验过程与原理，严谨有说服力。

五、参考文献

1. 数据库建表文件。lab2-3schema.sql
2. 数据文件。lab2-3data.sql