

基本信息

姓名：安志宏 求职意向：Java 后端开发

学历：本科 手机：15847883925

生日：1995.08.08 邮箱：2550364626@qq.com

毕业院校：中南民族大学 通信工程 2015.09-2019.06 现居地：内蒙古巴彦淖尔



工作经历

工作单位 起止时间 工作岗位

北京药帮忙科技有限公司 2021.03-2022.08 Java 后端开发工程师

深圳消安科技有限公司 2019.07-2020.08 嵌入式软件开发工程师

项目经历



工作单位 所在部门 项目名称 担任角色 起止时间

北京药帮忙科技有限公司 荷叶事业部-履约中心组 开放平台 Java 后端开发 2021.12-2022.08

**项目背景：**为提高订单履约率，增加获客率，节省与三方渠道对接成本，需要在原有开放平台的系统上设计一套对外提供的统一接口，供三方平台对接使用。目前已接入饿了么、拼多多、阳光保险等多家平台。 其中饿了么、拼多多日活在 2 千万单左右；

**项目架构：**Spring Boot + Mybatis + MySQL + Redis + RocketMQ + Elastic-Job**负责功能：**

1. **设计渠道对接方案:** 基于工厂、策略思想设计了开放平台对接渠道的代码框架。将渠道业务代码逻辑抽象分为订单、运单、问诊、药品、营销 5 个基本模块。将渠道对接开发改为配置化的方式，只需要在数据库中完成对应接口名配置即可完成渠道对接。无法通用的渠道只需要实现基本接口，实现对应渠道的逻辑即可。这样的对接方案不仅保证了业务代码的整洁，同时也将原本对接渠道开发时间需要 7 天缩减为只需要 1 天的研发时间，极大地提高了研发效率。

2. **饿了么渠道订单对接:** 主要实现了饿了么渠道 O2O 订单正反向业务流程。完成了渠道订单数据到我司订单数据的转换，保证我司订单接单时效。同时给结算业务提供了订单必要的金额信息，如：优惠金额计算、运费等，保证了结算等下游业务的正常运行。项目上线后饿了么日活达到 1 千万单左右，平均客单价达到 40 元左右。

**解决的问题：**

1. 商品表数据查询缓慢: 使用多线程优化下单流程，上线后通过 SkyWalking 发现饿了么回调接口耗时

长达 2s 左右。通过对调用节点的分析，发现因为查询商品映射表时耗时 2s 左右，然后当时统计

一下这张表的数据，大概在 1 千多万，并且查询的条件是全表扫描。最终加了联合索引解决了这个

问题。回调接口响应时间从 2s 减短了 1.2s，接口性能提升了 60%。

1. RocketMQ 消费队列消息积压: 线上同步百度商品信息的时候发现 RocketMQ 消息大量堆积。使用 FastThread 分析线程快照，发现线程池中的线程一直在等待 CountDownLatch 锁释放进一步析代码发现是因为在并发的操作下使用了不安全的集合对象，导致集合中出现了 null 对象。在线程池submit任务时，集合中对象获取自身属性导致空指针异常。但是这个异常被线程池 “吞”了，结果代码中 finally 释放锁的操作没有执行，导致线程阻塞，从而导致了消息积压。最终将非线程安全的集合替换为了线程安全的集合，设置了 await() 方法的超时时间，解决了 RocketMQ 消息队列阻塞的问题。
2. 订单数据幂等: 使用 Redis 实现分布式锁，解决饿了么下单、退单重推造成业务并发场景，导致出现

一单多同步、一单多退的问题。



个人技能

* 熟悉常用设计模式，熟练操作 MySQL 数据库，熟悉 MVCC 事务隔离机制，MySQL 索引优化
* 熟练使用 Java 常见的开源框架：Spring、Spring Boot、MyBatis；阅读过 Spring 源码，了解 Spring

的启动流程，Spring Bean的生命周期；了解 SpringBoot 自动装配原理；具备一定的 Spring 框架定

制开发能力；

* 熟悉高并发下多线程的开发，熟悉线程池的工作原理；了解 JDK 下 JUC 包实现原理；
* 了解 RocketMQ 消息堆积、消息重发、消息发送失败等场景的处理方法；
* 熟悉 Redis 缓存高并发场景，如缓存雪崩、缓存穿透、缓存击穿场景；
* 熟练使用 Linux、Git 常用的操作命令；



自我评价

1. 有责任感，学习能力强，有上进心。具有团队精神，善于与人沟通，表达能力良好

2. 个人账号：GitHub: https://github.com/an-1024/an-1024/blob/main/Contents.md