

基本信息

姓名：安志宏 求职意向：Java开发工程师

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**

南京亚信软件有限公司 2022.09-至今 Java后端开发工程师

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



专业技能

* 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 **Spring Boot** 自动装配原理；具备一定的 **Spring** 框架定

制开发能力；

* 熟悉 **JVM** 内存模型，熟悉 **JVM** 垃圾收集机制、**GC** 算法；了解 JVM 常用的配置参数；了解使用 **jinfo**、

**jstat** 等 **jdk** 提供的命令，**Arthas** 等性能监控工具对 **JVM** 进行参数调优；

* 熟悉高并发下多线程的开发，熟悉线程池的工作原理；
* 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；解决过线上 **RocketMQ** 消息

堆积的问题；

* 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；了解 **Redis** 线程模型，自己搭

建过 **Redis** 主从、哨兵等集群部署；

* 熟练使用 Linux、Git 常用的操作命令；



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 荷叶业务系统 Java 后端开发 2021.12-2022.08

**开放平台项目背景：**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**2000**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7天的研发时间缩减到2天，研发效率提高了**70%**。

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在400单左右，平均客单价达到40元左右。

**荷叶保险商城项目背景：**为扩展我司药品售卖渠道，扩大业务范围，我司决定与各大保险公司合作，将我司

的药品纳入保单中。因此需要开发一套荷叶保险商城的系统，具备配置险种、分佣结算、保单信息记录等

功能。该系统上线后，接入了阳光、泰康、众惠等保险公司。订单日活在**5000**单左右。

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

**负责功能：**

1. **保险渠道接入方案设计：**基于策略、模板方法思想，设计荷叶保险系统与三方保险公司的对接。按照保险公司的投保流程——记录投保人(受益人)等信息、核算保费、支付保费、生成保单；将投保流程抽象为一个公共父类，所有对应的保险渠道只需要继承这个父类实现父类中的方法即可。这样规范了保险渠道接入的开发流程，不同的渠道只需要实现自己的逻辑算法即可，相互之间没有业务干扰，便于日后代码的维护与扩展。同时减少了每次接入保险渠道的开发成本，由4人2天完成一个保险渠道对接，到后面2人2天完成一个保险渠道的对接，研发人员投入以及研发成本降低了**50%**。

**解决的问题：**

1. **RocketMQ 消息积压：**消费 RocketMQ 线程池中的线程异常，锁没有释放，最终导致消息积压。设置锁超时时间，线程池中捕获线程异常，最终解决问题；
2. **服务器崩溃：**大量商品查询请求打到服务器上，导致频繁 Full GC，最终导致服务器崩溃。添加热点商品缓存，调整 JVM 年轻代运行参数，扩大年轻代容量，使用 sentinel 限流，问题得以解决；



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>；