

XXX

xxxxxxxxxx | xxxxxxxxxxxx@qq.com | 陕西西安

23岁 | 男

研一在读学生

## 教育经历

沈阳化工大学

研一在读

大数据技术与工程 硕士 计算机学院

陕西理工大学

计算机科学与技术 本科 计算机学院

## 专业技能

- **计算机基础：**
  - 熟悉数据结构与算法。如链表、队列、栈、二叉树，以及快排、动态规划、回溯等算法。
  - 熟悉计算机网络，熟悉TCP/IP协议栈。如三次握手、四次挥手、流量控制、拥塞控制，以及HTTP协议。
  - 熟悉操作系统。如进程、线程、死锁、调度算法、内存分配、分页分段、虚拟内存、页面置换算法相关知识。
  - 熟悉常见的设计模式，如单例，工厂，代理、模板方法模式等。
- **Java：**
  - 熟练掌握Java基础知识，包括面向对象、异常处理、多线程、集合及其原理。
  - 熟悉Java8新特性，如Lambda表达式、Stream流、Optional类、时间日期API等。
  - 熟悉JVM。如内存结构、垃圾回收算法、垃圾回收器、类加载等。
  - 熟悉JUC。如线程池、ReentrantLock、JMM、CAS、AQS、ConcurrentHashMap、ThreadLocal等。
- **框架：**
  - 熟练掌握Spring、Spring MVC、Spring Boot、MyBatis、MyBatis Plus等主流开发框架。
  - 熟悉Spring，Spring MVC的基本原理，如IOC、AOP、Spring MVC执行流程等。
  - 熟悉Spring Cloud全家桶常见组件的使用，能独立开发微服务项目。了解熔断、降级、限流的使用。
- **数据库：**
  - 熟悉MySQL数据库以及常见优化手段，如索引、事务、锁、SQL优化、MVCC等。
  - 熟悉Redis的相关应用。如持久化方式、过期策略、缓存击穿、穿透、雪崩、分布式锁、分布式会话等。
- **中间件：**
  - 熟悉RabbitMQ的相关应用，如保证消息可靠性的confirm与return机制、死信队列、延时队列等。
  - 了解Elasticsearch和Nginx的基本使用，可用Nginx实现项目发布、反向代理、负载均衡等功能。
- **工具：**熟悉Git，Maven，了解Docker的基本使用。熟悉Linux操作系统及其常用命令。
- **前端：**熟悉HTML，CSS，JavaScript，Vue，Vuex，Vue Router，Element UI，Vant等前端知识。

## 实习经历

---

北京仁聚汇通信息科技有限责任公司杭州分公司（实习）

2022年07月 - 2022年08月

Java后端实习生

- **工作内容：**

- 开发公司内部低代码平台一些参数表的crud接口。
- 基于低代码平台，对上下游交互所需的来往报文进行转换以及配置。

- **实习收获：**

- 在开发上，熟悉了前后端分离项目的开发流程。
- 在代码风格上，特别注重注释的标注，代码的规范严格按照公司开发规范进行开发。
- 在沟通协作上，能够快速描述问题并协作同事进行处理。

## 项目经历

---

本地生活点评网

**技术选型：**SpringBoot，MybatisPlus，Redis，RabbitMQ，Vue

**项目描述：**该项目是一个进行本地生活信息分享、点评与交易的平台，主要实现了探店笔记发布，点赞排行榜，关注与取关，关注推送，签到与补签，抽奖，优惠券抢购，uv统计等功能。

- 使用**Token配合Redis实现分布式会话**，解决了集群下Session共享问题。基于**静态ThreadLocal**封装了线程隔离的全局上下文对象，便于在**请求内部读取用户信息**，减少用户查询次数。
- 采用Redis实现了**高频信息缓存**（分类信息，店铺信息），大大加快了请求响应速度，降低了20%的数据库访问压力。为了保证缓存和数据库的一致性，**缓存更新策略为先更新数据库，再删除缓存**，出现数据库与缓存不一致用超时删除来兜底。
- 为了解决**缓存击穿、穿透、雪崩**问题，**自定义缓存工具类**，业务中使用缓存时只需调用缓存工具类方法即可避免击穿、穿透、雪崩问题。
- 使用时间戳+序列号结合Redis实现**全局分布式id生成器**，保证**id全局唯一且递增**。Redis每天一key，避免无限增长超过 $2^{32}$ ，也方便统计每天、每月、每年的订单量。
- 使用redis+lua实现**可重入的分布式锁**，避免了集群下JVM锁失效的问题，解决了优惠券的超卖问题，实现了一人一单。
- 使用**RabbitMQ优化优惠券下单速度**，异步保存订单信息到数据库，接口的响应耗时由0.7s降低为110ms左右。
- 为了确保RabbitMQ消息的可靠性，**开启confirm和return机制**，确保消息能够到达队列。开启**消费者ack机制和retry机制**，确保消息能够被消费。使用**死信队列作为兜底方案**，消息多次消费失败进入死信队列进行人工处理。
- 使用RabbitMQ的**TTL机制模拟延时队列**，实现订单的超时取消。相比于定时任务轮询，**极大地减少数据库访问压力**。
- 使用Redis的**set实现关注与取关**，**取交集实现共同关注**，使用feed流的**推模式实现关注推送功能**。使用**RabbitMQ将推送异步化**，大大减少了接口的响应时间。为了避免普通分页读到重复数据，使用**滚动分页来获取推送消息**。
- 为了避免用户恶意刷评论，通过redis的**zset实现评论接口限流**，一小时只能评论30次。
- 使用Redis的**bitmap位图来存储用户签到与补签信息**，解决了使用数据库存储签到信息占用太多空间的问题，一人签到一个月只占用4字节。
- 使用Redis的**zset实现点赞排行榜**的功能。使用Redis的set实现简单的**优惠券抽奖**功能，用户连续签到即可抽奖。
- 使用Redis的**HyperLogLog进行每日页面uv统计**，合并一个月的数据，就能统计出每月uv。

## 商品交易平台

**技术选型：**SpringCloud，RabbitMQ，MybatisPlus，Redis，Redisson，MySQL，Vue

**项目描述：**该项目是一个商品交易平台，主要实现了商品搜索，购物车，下单，用户活跃统计等功能。

- 独立开发的网上交易平台，采用的**前后端分离架构**，提高了开发效率，减少了30%的开发时间。
- 采用了**统一的结果封装类**，规范了返回的结果。并且开发了**响应工具类**，使用静态方法提高了开发效率。
- 自定义统一的错误码，同时**封装了全局异常处理器**，达成了规范异常返回、屏蔽了项目冗余的报错细节。
- 为了**避免单点故障**问题，同时加快迭代速度，适应产品特性，因此后端由单体**Springboot应用重构为SpringCloud微服务应用**。解决了单体项目复杂度高的问题，使每个独立的服务能够更好地进行维护和拓展。
- 使用SpringCloud Gateway实现请求转发，在**网关层面解决跨域、用户鉴权**等问题，降低开发成本、提高安全性。
- 使用**Hystrix熔断器**实现订单**接口的降级和熔断**，从而保护整体服务的可用性，避免服务雪崩问题。
- 使用Redis的zset集合实现了**卖家评分排行榜**，使用Redis的bitmap位图实现了**活跃用户统计**。
- 采用**Redis实现了高频信息缓存**（首屏信息、分类信息），大大加快了请求响应速度，降低了20%的数据库访问压力。
- 使用**Redisson分布式锁解决商品超卖**问题，使用**RabbitMQ异步保存订单**到数据库，优化了接口的响应时间。使用**RabbitMQ的TTL机制来模拟延时队列**，实现了订单的超时取消功能。
- 前端采用**Nginx进行部署**，后端采用**Docker容器进行部署**。

## 成绩 / 荣誉 / 证书

---

- **成绩：**本科担任学习委员，学习成绩班级第一，年级前十。
- **荣誉：**优秀实习生 励志奖学金 三好学生
- **语言：**英语CET-6（595分）