



李晨

求职意向: java 软件工程师

个人简历

Personal Resume



湖南省郴州市



17773560880



lichen_stc@163.com

教育背景:

2018.9 - 2022.6 南华大学 软件工程 (本科)

主修课程:

C++程序设计、Java 程序设计、数据结构、编译原理、软件工程、操作系统原理、数据库系统原理、网络数据库、计算机网络技术

自我评价:

1. 本人性格开朗, 待人诚恳, 乐于助人, 不怕吃苦, 有很强的执行力和意志力
2. 学习能力强, 思路清晰, 善于从整体上分析、把握复杂事物, 对细节难题具备快速钻研解决能力;
3. 能快速地融入、凝聚团队;
4. 良好的表达与沟通能力, 积极主动, 对工作尽心尽责;
5. 对技术有热切的追求, 有不断学习的钻研精神

校园经历:

1. 大一、大二期间担任学院学生会成员, 并加入学院部门, 多次参与校园活动的管理工作
2. 课内技术掌握之余, 经常利用课外时间学习新知识, 如 Python、JVM 等。
3. 在校项目期间, 多次担任项目组长
4. 独立开发在线教育网址, 并上传到阿里云服务器上运行
5. 坚持写笔记做总结

专业技能:

- ✧ 熟练掌握 java 基础知识, 多线程, IO, 集合。熟悉 JVM, 有过 JVM 排查问题和调优的经验
- ✧ 熟悉 TCP/IP 网络协议
- ✧ 熟练掌握对MySQL, Oracle等关系数据库的使用, 对范式、隔离级别等有一定理解
- ✧ 熟练掌握HTML、CSS、JavaScript、JQuery、Ajax、Vue, Bootstrap, ECharts等前端集成UI框架技术

- ✧ 熟悉 Redis 原理和作为缓存的使用，熟悉 5 种基本类型的存储原理及使用场景，RDB 与 AOF 持久化的特点
- ✧ 熟练掌握 Mybatis 框架，理解 Mybatis 的底层原理
- ✧ 熟练掌握 Spring、SpringMVC、SpringBoot 框架，以及使用 SpringCloud 整合框架开发分布式服务
- ✧ 手写过 springMVC，spring IOC，AOP 框架，技术驾驭能力较强
- ✧ 熟悉 BIO，NIO，Netty 等常见 I/O 框架的使用，对 NIO 的底层实现有一定了解
- ✧ 熟悉 Linux 系统，熟悉其常见的操作指令，了解 shell 编程
- ✧ 熟悉 Git，Maven 等项目管理工具进行协同开发
- ✧ 熟练掌握数据结构和一些基本算法，熟悉代理，单例，工厂等常用的设计模式
- ✧ 熟悉 nginx，RabbitMq，docker，nacos 以及 eureka，ribbon，zuul，security 等框架
- ✧ 熟练使用 Idea 集成开发工具进行接口开发，有良好的编程风格
- ✧ 了解并使用阿里云服务进行接口开发，对 C++，Python 语言也有一定的了解
- ✧ 熟练使用 Idea 集成开发工具进行接口开发，有良好的编程风格

项目经验：

CRM 客户关系管理系统

软件环境：Apache Tomcat9.0/Mysql3.4.5/Windows10/JDK1.8

开发工具：IntelliJ IDEA

项目框架：Maven+Mybatis+jsp

项目描述：CRM 即客户关系管理，使用传统的 jsp+ajax 实现的 javaWeb 开发，用于管理与客户之间的关系。它的目标是通过提高客户的价值、满意度、赢利性和忠实度来缩减销售周期和销售成本、增加收入、寻找扩展业务所需的新的市场和渠道。

核心业务：

- 1，用户模块：简单的登录操作
- 2，市场活动模块：使用了条件查询和分页查询并展现市场活动，点击进入市场活动详细信息
- 3，线索模块（潜在客户）：关联和解除市场活动，执行线索转换的操作同时添加交易
- 4，交易模块：动态展现交易阶段内容及图标，点击阶段图标，更改交易阶段
- 5，统计图表：展现交易阶段统计图

技术要点：

- 1, 使用 Mybatis 框架对数据的 CRUD
- 2, 登入操作增加了 ip 地址的验证
- 3, 使用监听器在上下文域对象创建是将数据字典添加到服务器缓存中
- 4, 使用前端 UI 框架实现的分页, 日期, 自动补全等插件, 使用 ECharts 来展现统计图

项目总结:

- 1, 由于当时还没有学习 SpringMVC 的使用, 对于字段数较多的实体类的传输处理起来比较麻烦, 更深刻的感受到了 SpringMVC 的优势
- 2, 强化了对接口的开发流程的掌握

在线教育系统

软件环境: JDK1.8

开发工具: IntelliJ IDEA+Visual Studio Code

项目框架: 后端: SrpingBoot+SpringCloud+MyBatis-Plus+HttpClient+Maven+EasyExcel

前端: Node.js+Vue.js+element+NUXT+ECharts

中间件: Redis+nginx+nacos

项目描述: 在线交易系统, 分为前台网站系统和后台运营平台, 采用 B2C 模式, 前台用户系统包括讲师, 文章几大大部分, 使用了微服务技术架构, 前后端分离开发。

具体功能:

后台管理系统:

- 1, 管理员登录之后, 根据不同的管理员身份, 进行对应的管理操作
- 2, 对用户进行角色分配和菜单分配
- 3, 对讲师和课程进行添加, 修改, 删除操作, 并最终发布课程。
- 4, 生成统计数据, 并显示数据图表

前台用户系统:

- 1, 注册并登录进入主页面的动态 banner, 热门讲师和热门课程的显示
- 2, 点击名师导航进入名师列表的显示, 点击名师头像即可查看名师的详细信息
- 3, 点击课程导航进入课程列表的显示, 点击课程进入课程详细 (包括课程基本信息, 评论, 分类, 讲师, 课程大纲), 并判断课程是否需要购买, 如需要购买则点击购买跳转微信二维码付款, 成功后即可观看课程的相关视频

技术要点:

- 1, 在线教育项目采用前后端分离开发
- 2, 项目使用前端技术: 后台系统使用 vue-admin-template, 前台系统使用 Nuxt, 图表 Echarts
- 3, 后端技术: SpringBoot, SpringCloud: 项目中, 使用了阿里巴巴 Nacos 替代了 SpringCloud 一些组件, Nacos 作为注册中心和配置中心, Feign 用于服务调用, 服务之间的远程调用, 还使用了熔断器机制进行熔断降级, 后续使用了 Gateway 网关替代 nginx 解决跨域问题和负载均衡, Mybatis-plus 进行对数据库的操作。使用 Swagger 生成接口文档, 使用 Git 进行代码的管理。最终使用 Docker+Jenkins 进行打包部署。
- 4, 前台系统为了安全采用了单点登录系统使用了 jwt, 使用 redis 进行存储用户的 token 来模拟 session 机制
- 5, 后台系统使用 Spring Security 权限框架分配给不同登录用户的角色
- 6, 使用 echarts 完成播放量, 订单等不同维度的图表显示
- 7, 使用 Redis 对主页面的信息进行缓存

- 8, 阿里云第三方短信实现注册, 微信扫码登录, 使用微信支付, 生成支付二维码, 微信最终支付
- 9, 使用阿里云 OSS 进行视频和图片的存储, 并使用了阿里云视频点播进行视频在线播放

项目总结:

- 1, 多次路由跳转到同一个页面, 页面中 created 方法只会执行一次。解决方案: 使用 vue 监听 Nodejs 不能直接运行 ES6 模块化代码, 需要使用 Babel 把 ES6 代码转换为 ES5 代码执行
- 2, 项目中请求出现跨域问题。一开始是通过在 Controller 上添加跨域注解, 后来使用网关解决了, 但使用网关之后再次出现跨域问题, 查阅资料后, 发现注解加网关进行了两次跨域请求, 把注解去掉即可
- 3, 在上传视频时, 出现 403 错误, 查阅资料发现是 Nginx 有上传文件打下限制, 通过配置解决
- 4, Maven 加载项目时默认不会加载 src-java 文件夹中的 xml 类型文件, 通过添加配置解决
- 5, 分布式 id 生成器在前端无法处理, 总是在后三位进行四舍五入, 查阅资料后, 得知分布式 id 生成器的 id 是 19 个字符的长度, 前端 js 脚本对整数的处理能力只有 2 的 53 次方, 也就是最多只能处理 16 个字符, 最终的解决方案是把 id 在程序中设置为字符串

零食在线商城

软件环境: JDK1.8

开发工具: IntelliJ IDEA

项目框架: SpringBoot+SpringCloud+MyBatis+Vue

项目描述: 该商城是一个零食类的电商购物网站 (B2C), 用户可以在线购买商品, 加入购物车, 下单, 付款的购物流程

具体功能:

用户功能:

- 1, 新用户的登录注册, 邮箱验证注册, 登录及身份验证
- 2, 首页展现可根据商品的类别查询和名字模糊查询使用, 在轮播图界面有相关推荐商品展现
- 3, 商品详细信息展现, 该页面包括商品的详细描述, 价格库存等详细信息
- 4, 商品购买模块, 用户选择商品进行立即购买或加入购物车的操作, 进入购物车点击结算, 跳转到订单页面选择或添加收货信息确认订单信息后支付
- 5, 支付使用的是支付宝的沙箱支付来测试模拟支付
- 6, 我的订单模块展现订单的状态, 个人信息模块可修改个人信息

后台管理:

- 1, 管理员登录, 不允许同一用户在多处登录
- 2, 对商品进行上架, 下架, 商品类型添加, 商品数据的可视化管理
- 3, 对客户的订单进行发货等操作

技术要点:

- 1, 该系统是使用 eureka 注册中心进行服务的注册, zuul 作为微服务网关, ribbon 作为负载均衡服务器, config 作为配置中心实现的微服务架构。
- 2, 使用 Feign 进行服务之间的远程调用, 并使用熔断降级机制
- 3, 利用消息中间件 RabbitMQ, 对项目进行配置的动态修改
- 4, Zipkin 链路追踪, 通过 UI 组件帮助寻找系统中出现的延迟升高问题及系统性能瓶颈根源
- 5, Websocket 实现后台的点点登录

项目总结:

1, 项目迁移到 Spring-Cloud 的时候, 经过网关时, 前端传递的 cookie 后端一直获取不了, 看了 cloud 中 zuul 的源码, 发现向下游传递数据的时候, zuul 默认过滤了敏感信息, 将 cookie 过滤掉了, 解决的方案是在配置文件中将请求头的过滤器清除掉, 使 cookie 可以向下游传递