

项目推荐

项目说明

对于很多工科专业来说，尤其是计算机专业的同学来说，项目经验是衡量你是否符合公司要求的一个重要考虑因素，并且简历上很大一部分占比也是需要项目经验支撑的。一个跟你投递公司相符合的项目也更能为你加分，更容易赢得进入公司的门票。

优秀的项目

一个好的项目主要有以下几点：

1. 项目拥有前沿的，主流的技术栈；
2. 能够生产上线的，实际投产的项目；
3. 项目是能够解决部分问题的，项目的架构和项目的技术栈是为了解决业务需求而设计的，有针对性的；
4. 项目架构和代码规范，可读性高，可扩展性强；
5. 项目具有规范的文档；

项目等级

实习项目 > 上线部署项目 > 个人学习项目 > 简单课设项目

分布式，微服务 > 单体项目

前后端分离项目 > 前后端不分离项目

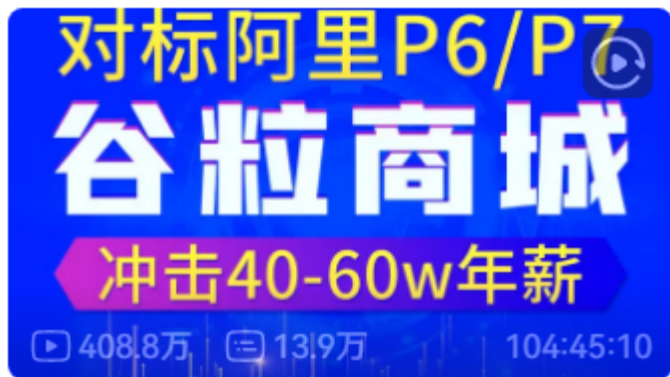
业务类项目

业务类项目主要是指对相关业务进行增删改查的项目，本质上都是增删改查，但是区别在于：

1. 对于业务的理解，复杂的业务会涉及到很多内容，比如不同系统之间的交互，业务逻辑之间如何解耦等；
2. 项目的系统架构和设计，比如最简答的单体项目，分布式，微服务，DDD 等；
3. 项目用到的技术栈以及技术选型，比如消息队列如何选择，Kakfa，RocketMQ 等；
4. 项目遇到的难点，如何解决，比如如何保障项目高可用等；

谷粒商城

链接：[🔗 Java 项目《谷粒商城》Java 架构师 | 微服务 | 大型电商项目_哔哩哔哩_bilibili](#)



Java项目《谷粒商城》Java架构师 |
微服务 | 大型电商项目

UP 尚硅谷 · 2020-3-31

优点

1. 技术栈丰富和主流，基本上 Java 开发涉及到的技术栈均有设计；
2. 视频讲解详细，一步一步带你做，适合没有毫无经验的小白手把手教学；
3. 配套的资料丰富，比如详细的文档，做过的人的踩坑经验，涉及到的面试题，如何包装等；
4. 商城项目是很多公司都会涉及到业务，匹配度较高；
5. 头部培训机构出品，经过了很多学员以及自学者的验证；
6. 有部署上线的教学，一整条龙服务；

缺点

1. 流传度高就意味着烂大街，很多应届生都用的这个项目；
2. 容易让人导致你是培训班出来的；
3. 项目环境配置麻烦，经常会卡在某个环境；
4. 2020 年的项目很多包以及技术栈版本都更新了；
5. 项目内部模块繁多，业务相对复杂，做完需要较长的时间；

牛客网论坛

链接：🔗 [课程列表 牛客网 \(nowcoder.com\)](https://www.nowcoder.com)

技术栈：

1. Spring
2. SpringMVC

- 3. SpringBoot
- 4. MySQL
- 5. Redis
- 6. KafKa
- 7. Elasticsearch
- 8. Spring Security、Spring Actuator

牛客

课程首页 / 高薪求职项目课

教室

作业/测试/成绩

提问/答疑/交流

课程说明

0%

学习进度

0%

作业完成度

0.0

测试平均分数

周四03-02

已打卡0天
排名0

今日打卡

第1章 初识Spring Boot，开发社区首页

课程学习

课堂练习

课后作业

0%

0.0%

未完成

课程学习

1.1 课程介绍 免费

1.2 搭建开发环境 免费

课后作业

1.3 课后作业1 免费

1.4 课后作业2 免费

1.5 课后作业3 免费

课程学习

1.6 Spring入门 免费

1.7 随堂练习1 免费

1.8 随堂练习2 免费

优点:

1. 相对于谷粒商城来说，业务逻辑和模块更为简单，更适合零基础的同学；
2. 配套视频讲解详细，同样包含部署讲解；
3. 并且包含了开发场景中的单元测试，项目监控等；
4. 课程最后对项目进行了总结，包括总体回顾和面试题分析；
5. 配套的资料丰富，比如详细的文档，做过的人的踩坑经验，涉及到的面试题，如何包装等；
6. 整个项目课程全部免费；

缺点：

1. 和谷粒商城一样，流传度太高了，同样面临着烂大街的困境；
2. 项目没有明显的难点，不够出彩，需要适度包装并改进；
3. 项目没有明显的亮点，技术栈的使用不够深入；
4. 单体项目，可以适度改造成分布式，或者自行添加相关功能；

分布式抽奖系统

链接：<https://gitcode.net/KnowledgePlanet>

视频说明：[🔗 Lottery 抽奖秒杀系统 - 基于领域驱动设计的四层架构实践 哔哩哔哩 bilibili](#)

简介：Lottery 抽奖系统 项目是一款互联网面向 C 端人群营销活动类的抽奖系统，可以提供抽奖活动玩法策略的创建、参与、记账、发奖等逻辑功能。在使用的过程中运营人员通过创建概率类奖品的抽奖玩法，对用户进行拉新、促活、留存，通常这样的系统会用在电商、外卖、出行、公众号运营等各类场景中。

优点:

1. 星球中配套有详细视频和文档说明, 已经许多星球里的面试经验;
2. 星球主小傅哥是一线大厂高级技术大牛, 并且能及时回答你的各种技术问题, 非常有耐心;
3. 比较小众, 撞车概率少;
4. 业务复杂程度适中, 并且技术栈和架构思想非常主流;
5. 整个文档采用渐进式开发, 能够一步一个脚印带领大家做;
6. 亮点多: DDD 设计, 分布式, 任务调度, 最终一致性等;
7. 代码结构漂亮, 涉及到多种设计模式的实际应用, 各种接口与抽象能够让大家感受到开发的魅力;

缺点:

1. 虽然是渐进式开发文档, 但是不像谷粒商城等完全手把手带你做, 需要一定的基础;
2. 代码风格成熟且优美, 并且多种设计模式的应用会让部分新手不太理解, 消化需要一定的时间;
3. 项目不免费 (为知识付费很值得), 具体你们可以进小傅哥的主页看看。👉 [小傅哥の码场的个人空间_哔哩哔哩_bilibili](#)

IM 通讯系统

链接: 🔥 [Netty+SpringBoot 开发即时通讯系统 - 慕课网\(imooc.com\)](#)

仿 B 站视频网站项目

链接: 🔥 [SpringBoot 2.x 实战 SpringBoot 2.x 教程 - 慕课网\(imooc.com\)](#)

没有前端

秒杀项目

优点:

1. 资料丰富，无论是免费还是收费的，各种秒杀系统网上都很多；
2. 亮点多，涉及到高并发，高可用等互联网领域的常见亮点；
3. 技术选型多，无论是单体秒杀还是分布式秒杀各种实现都很多；

缺点

1. 相对来说比较常见，流传度比较广；
2. 业务逻辑并不复杂，但是需要足够深入，不然容易被问倒，面试问题是一环扣一环的；
3. 由于学生个人设计到的场景中很难实际接触秒杀业务，很多时候无法真正模拟业务场景；

轮子类项目

造轮子的意思就是造工具，造框架，设计到的少量业务，本质上是对工具功能的设计以及开发，对于传统的增删改查来说，有一定的难度，需要对轮子有比较深入的认知和分析，代码通常来说比较抽象，需要好好琢磨消化。

手写 Spring

Spring 框架是 Java 开发中必不可少的一部分，也是每个 JavaBoy 必须要会的框架，也是 Java 生态的核心和关键点

链接：[🔗 2022 年：《手写 Spring：渐进式源码实践》 | bugstack 虫洞栈](#)

优点:

1. 通过手写 Spring 能够更加好的学习 Spring 的核心思想 IOC 和 AOP;
2. 能够学习到一些设计模式的应用;
3. 相比于简单的增删改查对于技术的提高更有效果;
4. 能够通过引导将面试问题引导到关于 Spring 的相关问题上;

缺点

1. 手写 Spring 通常只是实现其核心功能, 不可能像 Spring 框架那样完善, 需要适当的引导面试官, 不然容易导致面试官不想提问;
2. 相比于业务类, 上手难度更高一点;

手写 RPC

rpc 是指远程调用, 通过 rpc 能够使调用远程项目像调用本地项目一样简单

技术栈: 通信框架 Netty, 序列化, 动态代理等;

链接:  [我手写了一个 RPC 框架。成功帮助读者斩获字节、阿里等大厂 offer。 - JavaGuide - 博客园 \(cnblogs.com\)](#)

优点

1. 跟所有轮子类项目一样, 能够帮助我们更好的理解 RPC 框架;
2. 项目设计到 Netty 的使用, 能够帮助大家了解更多网络通信底层知识, 这也是面试常考的点;
3. 市面上资料丰富, 面试题更是一搜就有, 准备相对轻松;

缺点

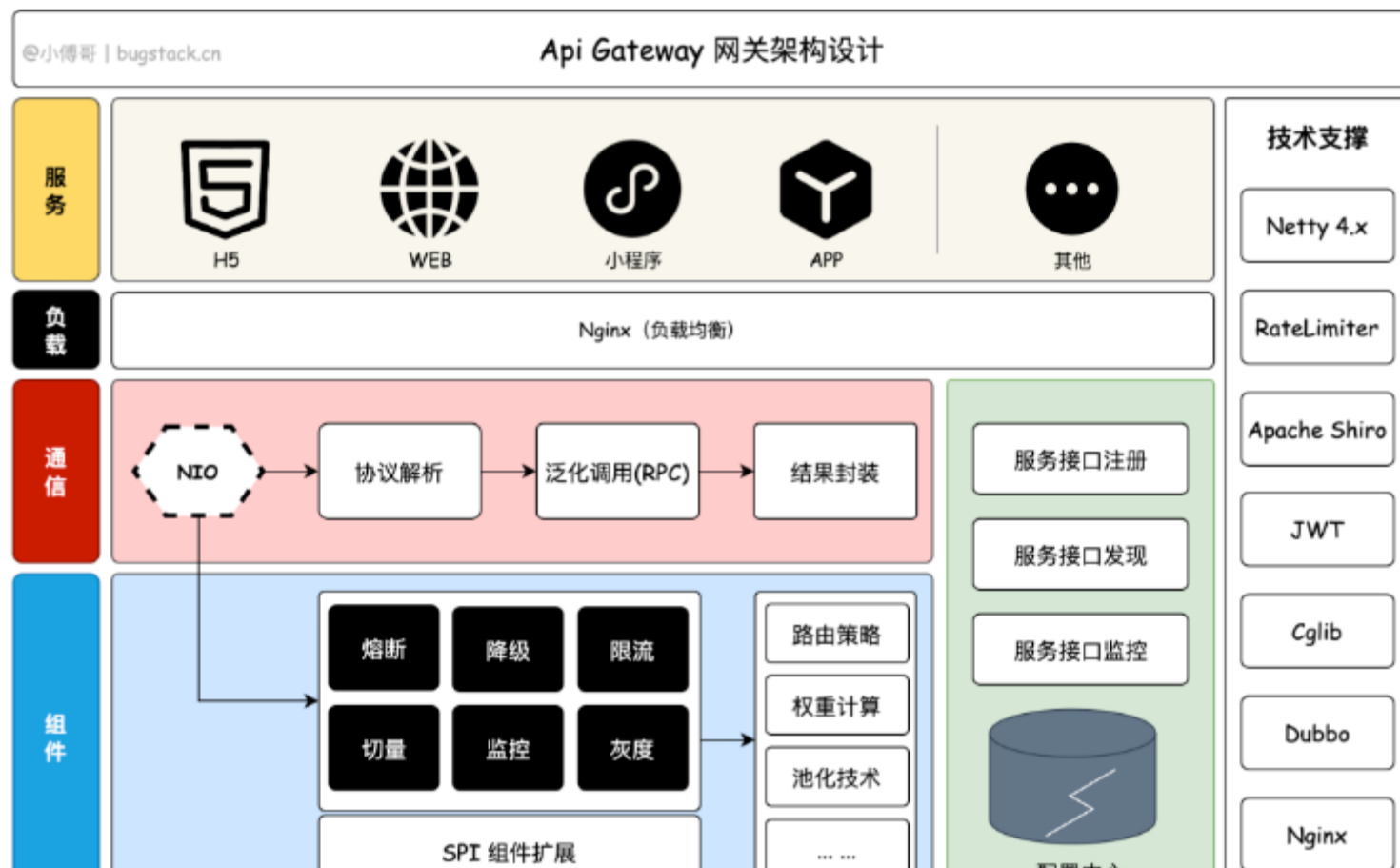
1. 前两年是非常好的轮子项目, 21 年 22 年用的人太多了, 相对来说是轮子里的烂大街;

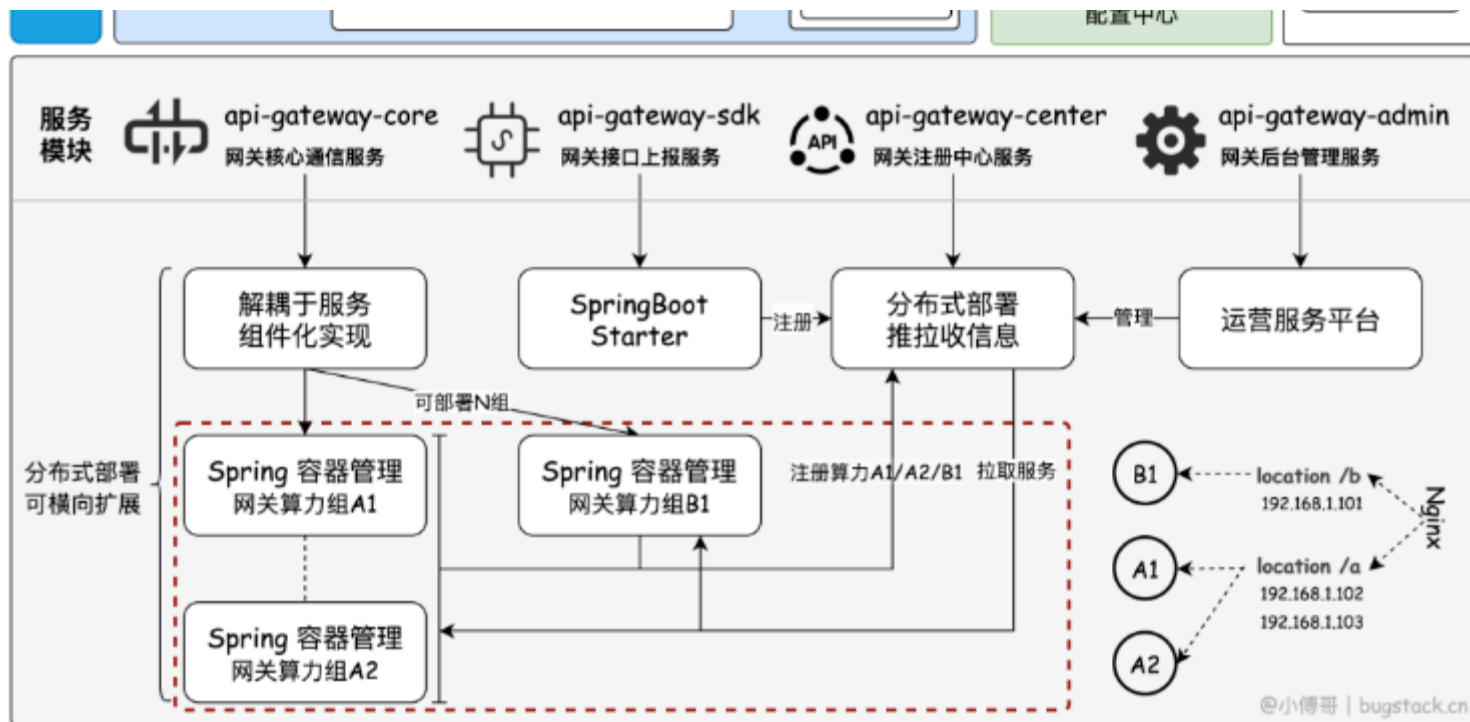
手写 API 网关

网关（Gateway）是转发其他服务器通信数据的服务器，接收从客户端发送来的请求时，它就像自己拥有资源的源服务器一样对请求进行处理。而 API 网关也是随着对传统庞大的单体应用（All in one）拆分为众多的微服务（Microservice）以后，所引入的统一通信管理系统。用于运行在外部 http 请求与内部 rpc 服务之间的一个流量入口，实现对外部请求的协议转换、参数校验、鉴权、切量、熔断、限流、监控、风控等各类共性的通用服务。

连接：[👉 如果让我设计一套，TPS 百万级的《API 网关》！（qq.com）](#)

架构图





优点

1. 星球中配套有详细视频和文档说明，已经许多星球里的面试经验；
2. 星球主小傅哥是一线大厂高级技术大牛，并且能及时回答你的各种技术问题，非常有耐心；
3. 比较小众，撞车概率少；
4. 整个文档采用渐进式开发，能够一步一个脚印带领大家做；
5. 代码结构漂亮；

缺点

1. 难度较大，需要较长时间消化和理解；
2. 目前还没更新完毕，但是核心章节已经更完了；

3. 非免费项目，需要为知识付费；

项目包装

项目描述：

1. 涉及到的技术栈；
2. 项目的简单介绍；
3. 个人贡献；

项目包装

1. 突出项目的重点和难点，比如常见的高并发，高可用等；
2. 突出你如何解决问题，比如内存溢出如何解决，消息堆积如何解决，如何防止超卖等；
3. 结合实际适当修改项目的应用场景；
4. 实习期间的项目要学会“拿来主义”；
5. 没有难点，没有亮点就要自己虚构场景来生成亮点；
6. 上线了的项目一定要了解 QPS 等参数，服务器配置等信息；

项目引导

1. 优秀的面试者会把面试官往自己熟悉的领域带，项目经验同样如此；
2. 在自我介绍和简历的项目描述中都应该适当引导面试官；

