

2023

夏季大厂版

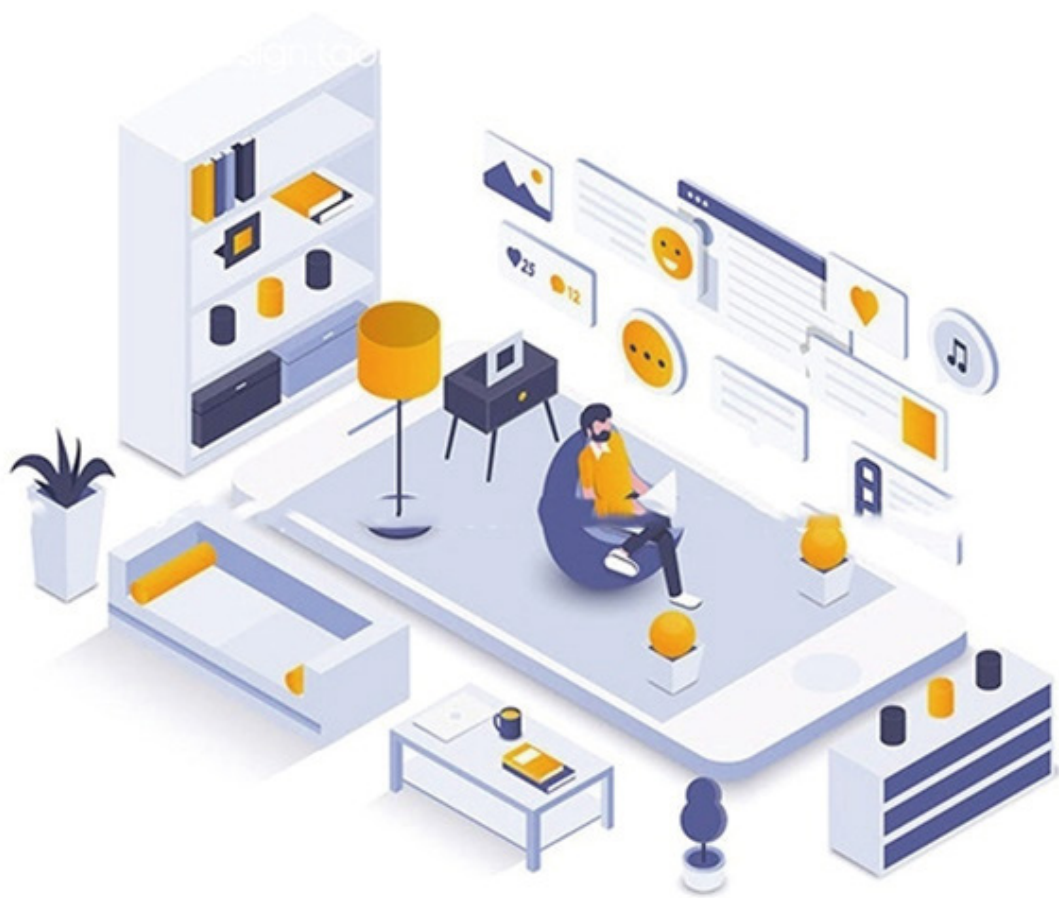
Java大厂架构师 训练营大纲

进阶架构师
直达阿里P7



课程简介

课程主要面向 1-5 年工作经验的 Java 后端工程师，以及 Java 后端技术基础较好的应届毕业生，教研团队均为在职 BAT/TMD 等一线大厂资深技术专家及一线面试官，根据自身在大厂多年积累的实战经验、大厂的用人需求及大厂的内训体系，整理出全套符合一线大厂用人需求的学习路径，通过线上学习 + 大厂真实项目 + 在线答疑 + 简历指导 + 面试官直推等内容，帮助同学在 6 个月内快速进阶阿里 P7、百度 T6。



适宜人群



只会使用基础框架，难以突破 Java 后端核心技术瓶颈



实战项目经验 + 底层源码薄弱，长期作为 crud 程序员，技术进步缓慢，求职面试无优势



希望进入 BAT TMD 等互联网一线大厂，但不知如何做到



希望找到最专业的机构学习，不想被劣质机构误导



薪资低于 30K，希望通过系统学习涨薪 50% 以上



希望有人日常监督学习，解决自律性

为什么选择爪哇

目前国内**唯一**一家授课教师均为
在职阿里、字节、百度等一线大
厂顶级 Java 后端大牛（阿里 P8、
百度 T7），全程亲自带，大厂内
部导师制教学

内容覆盖所有 Java 后端主流技
术栈，**实战项目均为个一线大厂
真实项目**，每个教学模块一一
对应各大厂的招聘要求



承诺内推覆盖所有国内一线大
厂，且为**面试官直推**，简历直达
各部门技术 TL

承诺为同学们提供不少于 20% 的
涨薪幅度，**签署薪资保障合同**，
对课程质量以学习成果负责

优秀师资介绍

以下为部分授课教师，具体以实际安排为主



缪斯 (muse) 百度资深研发工程师/前京东、阿里技术专家

目前任职于百度，T8级资深研发工程师

曾就职于用友、搜狐视频、蚂蚁金服、京东及百度等多家互联网龙头企业，涉及商用CRM、支付、物流、云、O2O业务开发经验



诺亚 阿里巴巴达摩院后端资深后端专家

曾就职于 菜鸟网络

对物流系统、自动驾驶数据处理分析、大规模实时离线计算 有深入的理解
在服务稳定性 大数据处理 领域建设有丰富的经验

曾经参与过日活上亿级别的物流分单系统开发以及亿级别数据处理系统的研发和维护



Tim 阿里巴巴P8高级技术专家

目前就职于阿里巴巴

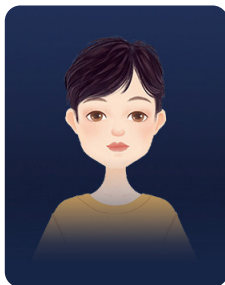
现役阿里巴巴P8高级技术专家，Java资深架构师，曾负责淘宝直播数据中台的构建，参与并负责多个大型项目，精通JavaEE多层体系架构软件设计与开发，拥有丰富的高并发实战经验。



Neo 百度资深开发工程师

主要从事后端开发，大数据开发

擅长分布式，大数据领域的相关技术栈。曾对Spark/Hadoop等开源软件进行过源码级别的开发、调优。参与过集团级别的大型数据中台建设，对数据查询，数据流通等数据治理领域项目有一线的开发设计经验。



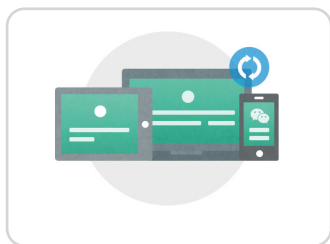
玉峰 字节跳动资深构架师/前阿里巴巴java领域高级专家

爪哇教育java课程教学教研负责人

目前就职于字节跳动，资深架构师，前阿里巴巴P8级员工。8年工作经验，曾就职于腾讯、百度、阿里等一线互联网大厂，负责过腾讯云数据平台、凤巢分析平台及淘宝数仓项目

精通javaEE多层体系架构软件设计与开发，拥有丰富的高并发实战经验

全程服务 放心托付

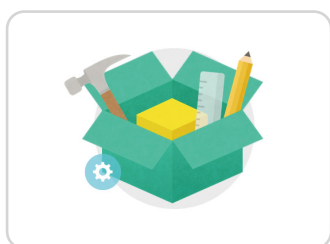
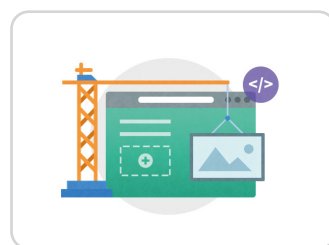


线上学习

一线技术及时更新 与大厂老师互动教学

大厂项目实战

独家一线大厂项目，从大厂设计理念到项目落地 补强核心项目短板

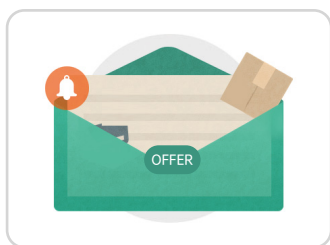
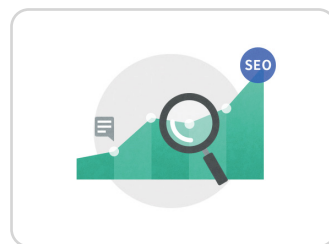


陪同学习

班主任全程陪伴 在线答疑 保障学习效果

资源共享

学习资料和数据源码免费发放 定时更新

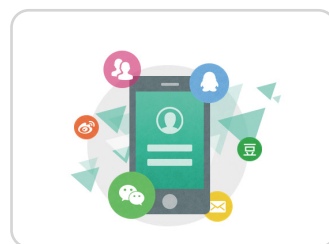


简历指导

对应大厂标准 提出优化意见 打造高分简历

无限内推

覆盖国内任意一线大厂 面试官直推 保障就业





同学欢迎你

来到训练营

- ◆ Java后端开发者必会知识点
- ◆ Java后端发展路径
- ◆ 训练营整体概述



01 实力筑基

模块 目标

- 01 扎实的java编程基础
- 02 具备良好的识别和设计通用框架及模块的能力
- 03 具备高度的抽象设计能力，具备独立的分析和设计实现能力，熟练应用常用的设计模式

设计模式

- 1.1.1 六大设计原则
 - 1.1.1.1 开闭原则
 - 1.1.1.2 依赖倒置原则
 - 1.1.1.3 单一职责原则
 - 1.1.1.4 接口隔离原则
 - 1.1.1.5 迪米特法则
 - 1.1.1.6 里氏替换原则
- 1.1.2 高频设计模式
 - 1.1.2.1 工厂设计模式
 - 1.1.2.2 策略设计模式
 - 1.1.2.3 单例模式
 - 1.1.2.4 命令模式
 - 1.1.2.5 代理模式
 - 1.1.2.6 模板模式
 - 1.1.2.7 复合模式实战

反射

- 1.2.1 反射机制
- 1.2.2 内置Class实例和数组的生命周期
- 1.2.3 反射实战
- 1.2.4 反射与设计模式结合

泛型

- 1.3.1 Java自动装箱拆箱
- 1.3.2 泛型的使用
- 1.3.3 泛型的类型擦除

容器

- 1.4.1 List
- 1.4.2 Map
- 1.4.3 Queue

02 Spring

模块 目标

- 01 熟练掌握目前流行开源框架，并深入理解其核心思想、实现原理
- 02 重点理解IOC、AOP、注解原理

IOC与AOP

IOC

BeanFactory

ApplicationContext

Bean的作用域

Bean的生命周期

AOP

AOP原理，代理

声明Pointcut

声明Advice

通知参数

02 Spring

模块 目标

- 01 熟练掌握目前流行开源框架，并深入理解其核心思想、实现原理
- 02 重点理解IOC、AOP、注解原理

springboot 原理与实战

- 1.5.3.1 Spring事务传播原理
- 1.5.3.2 请求参数绑定
- 1.5.3.3 拦截器Inteceptor使用
- 1.5.3.4 常用注解
- 1.5.3.5 自定义DispatcherServlet
- 1.5.3.6 全局异常处理器
- 1.5.3.7 Spring启动与请求过程
- 1.6.1 SpringBoot启动原理
- 1.6.2 嵌入式web容器
- 1.6.3 自动装配注解
- 1.6.4 Spring注解驱动设计模式
- 1.6.5 Spring5常用注解
- 1.6.6 集成 Mybatis plus
- 1.6.7 集成 Spring Data JPA
- 1.6.8 多数据源配置
- 1.6.9 集成Swagger
- 1.6.10 单元测试

03 SQL

模块 目标

- 01 通mysql的开发设计和调优，了解存储引擎，对SQL查询性能优化有深刻了解
- 02 熟悉分库分表，了解数据库原理

MyBatis架构与原理

- 1.7.1 MyBatis与Hibernate
- 1.7.2 MyBatis架构
- 1.7.3 Mapper的动态代理
- 1.7.4 执行器Executor
- 1.7.5 数据库会话器StatementHandler
- 1.7.6 通过StatementHandler看执行细节
- 1.7.7 参数处理器ParameterHandler
- 1.7.8 ResultSetHandler

事务与锁

- 1.8.1 事务ACID特性
- 1.8.2 事务并发引起的问题
 - 1.8.2.1 DirtyRead
 - 1.8.2.2 Unrepeatable Read
 - 1.8.2.3 Phantom Read
 - 1.8.2.4 JDBC解决方法：事务隔离级别
 - 1.8.2.5 死锁

索引原理

- 1.8.3.1 B-Tree索引
- 1.8.3.2 主键索引
- 1.8.3.3 辅助索引
- 1.8.3.4 组合索引

03 SQL

模块 目标

- 01 通mysql的开发设计和调优，了解存储引擎，对SQL查询性能优化有深刻了解
- 02 熟悉分库分表，了解数据库原理

SQL执行原理

- 1.8.4.1 SQL执行原理
- 1.8.4.2 慢SQL分析
- 1.8.4.3 慢SQL优化
- 1.8.4.4 分库分表
 - 1.8.4.4.1 垂直拆分
 - 1.8.4.4.2 水平拆分
 - 1.8.4.4.3 分库分表带来的问题
 - 1.8.4.4.4 什么时候考虑切分
 - 1.8.4.4.5 案例分析

分库分表原理与实战

- 1.8.4.4.6 MyCat
 - 配置详解
 - 分片规则详解
 - 管理命令
 - 读写分离
 - 故障切换
 - 高可用配置
 - 性能优化
 - MyCat总体架构
 - 内存管理与缓存架构
 - 多数据库支持
 - 核心原理
- 1.8.4.4.7 实战：SpringBoot+Mycat+MySQL实现分表分库

04 JVM

模块 目标

- 01 精通JVM调优、熟悉内存模型、了解常见JVM垃圾收集算法

JVM内存模型 垃圾回收算法 调优技巧 常见面试题

- 1.9.1 JVM内存模型
- 1.9.2 内存分配策略
- 1.9.3 对象头
- 1.9.4 类加载
 - 1.9.4.1 类加载机制
 - 1.9.4.2 双亲委派模型
 - 1.9.4.3 自定义类加载器
 - 1.9.4.4 线程上下文类加载器
 - 1.9.4.5 Tomcat类加载结构
- 1.9.5 判断对象存活的算法
- 1.9.6 垃圾回收算法
- 1.9.7 垃圾回收器
- 1.9.8 常用JVM参数
- 1.9.9 对象的强软弱虚引用
- 1.9.10 常见OOM及原因
 - 1.9.10.1 堆溢出
 - 1.9.10.2 栈溢出
 - 1.9.10.3 方法区溢出
- 1.9.11 JVM常用监控命令
 - 1.9.11.1 Jps
 - 1.9.11.2 Jstate
 - 1.9.11.3 Jmap
 - 1.9.11.4 Jstack
- 1.9.12 CPU或内存占用过高排查方案
- 1.9.13 tomcat优化配置

05 JDK8-21

模块 目标

- 01 JDK8-21新特性
- 02 深入了解JDK8
- 03 爆火的JDK17
- 04 提升编程实战能力

各版本特性汇总

接口私有方法 (JDK9)
本地变量类型推断 (JDK10)
HTTP Client (JDK11)
Switch 表达式 (JDK14)
文本块 (JDK15)
instanceof 的模式匹配 (JDK16)
Record (JDK16)
Sealed Class (JDK17)

函数式编程

1.10.1 函数式编程基础、函数式编程API
1.10.2 curry,composite,transducer...
1.10.3 functor,monad
1.10.4 lambda演算及运算符

JDK新特性详解

Java模块化
ZGC垃圾回收
诊断和监控
字符串压缩
Java flow API
新一代JIT编译器 Graal

06 消息中间件

模块 目标

- 01 熟悉分布式消息队列并了解其原理
- 02 熟练掌握底层中间件
- 03 精通分布式事务并对其原理有深入理解

RocketMQ

- 2.1.1 RocketMQ
 - 2.1.1.1 Rocketmq、kafka、Rabbitmq的详细对比
 - 2.1.1.2 RocketMQ集群概述
 - 2.1.1.2.1 Name Server
 - 2.1.1.2.2 Broker
 - 2.1.1.2.3 Consumer
 - 2.1.1.3 各种消息
 - 2.1.1.3.1 同步消息
 - 2.1.1.3.2 异步消息
 - 2.1.1.3.3 单向消息
 - 2.1.1.3.4 顺序消息
 - 2.1.1.3.5 批量消息
 - 2.1.1.3.6 过滤消息
 - 2.1.1.3.7 事务消息
 - 2.1.1.4 高级功能
 - 2.1.1.4.1 负载均衡
 - 2.1.1.4.2 消息的同步和异步
 - 2.1.1.4.3 消息重试
 - 2.1.1.4.4 消息队列
 - 2.1.1.4.5 消息幂等
 - 2.1.1.5 消息存储
 - 2.1.1.5.1 commitLog
 - 2.1.1.5.2 ConsumeQueue
 - 2.1.1.5.3 消息存储方式
 - 2.1.1.5.4 消息订阅
 - 2.1.1.6 Producer与Consumer实战

06 消息中间件

模块 目标

- 01 熟悉分布式消息队列并了解其原理
- 02 熟练掌握底层中间件
- 03 精通分布式事务并对其原理有深入理解

kafka

2.1.2 kafka

2.1.2.1 什么是消息系统?

2.1.2.2 Kafka的定位

2.1.2.3 Kafka的特性与使用场景

2.1.2.4 Kafka 安装与使用

2.1.2.5 Kafka核心概念

2.1.2.5.1 Broker

2.1.2.5.2 Consumer Group

2.1.2.5.3 Topic

2.1.2.5.4 分区 (Partition)

2.1.2.5.5 副本 (Replica)

2.1.2.5.6 分区平衡

2.1.2.5.7 Partition的读和写

2.1.2.6 分区

2.1.2.6.1 副本机制

2.1.2.6.2 Leader选举

2.1.2.6.3 分区重新分配

2.1.2.6.4 自动再均衡

2.1.2.6.5 修改分区副本、修改副本因子

2.1.2.6.6 分区分配策略

RangeAssignor

RoundRobinAssignor

StickyAssignor

自定义分配策略

2.1.2.6.7 物理存储

2.1.2.6.8 消息可靠性保证

2.1.2.6.9 消息一致性保证

2.1.2.6.10 消息重复的场景及解决方案

07 分布式缓存

模块 目标

- 01 熟悉Redis使用及原理
- 02 精通分布式锁并对其原理有深入理解

Redis原理与实战 分布式锁 其他常见缓存

2.2.1 什么是缓存、为什么使用缓存

2.2.2 缓存分类

2.2.2.1 客户端缓存

2.2.2.1.1 页面缓存

2.2.2.1.2 浏览器缓存

2.2.2.1.3 APP上的缓存

2.2.2.2 网络中的缓存

2.2.2.2.1 web代理缓存

2.2.2.2.2 边缘缓存

2.2.2.3 服务端缓存

2.2.2.3.1 数据库缓存

2.2.2.3.2 平台级缓存

2.2.2.3.3 应用级缓存

2.2.3 Ehcache与Guava Cache

2.2.3.1 Ehcache

2.2.3.1.1 核心概念

2.2.3.1.2 基本使用

2.2.3.1.3 数据缓存策略

2.2.3.1.4 Ehcache架构图

2.2.3.1.5 ehcache spring boot

2.2.3.2 Guava Cache

2.2.3.2.1 Guava Cache的适用场景

2.2.3.2.2 Guava Cache的创建方式

2.2.3.2.3 缓存数据删除

2.2.3.2.4 并发场景下的使用

2.2.4 集中式缓存Memcached

2.2.4.1 核心概念

07 分布式缓存

模块 目标

- 01 熟悉Redis使用及原理
- 02 精通分布式锁并对其原理有深入理解

Redis原理与实战 分布式锁 其他常见缓存

- 2.2.4.2 常用命令
 - 2.2.4.2.1 存储命令
 - 2.2.4.2.2 查找命令
 - 2.2.4.2.3 统计命令
- 2.2.4.3 Memcached特征
- 2.2.4.4 Memcached的一些问题
- 2.2.4.5 Memcached内存存储
 - 2.2.4.5.1 Slab Allocation机制
 - 2.2.4.5.2 使用 Growth Factor进行调优
 - 2.2.4.5.3 Item
- 2.2.4.6 典型问题解析
 - 2.2.4.6.1 过期机制
 - 2.2.4.6.2 哈希算法
 - 2.2.4.6.3 热点问题
- 2.2.4.7 Spring Boot 集成Memcached
- 2.2.4.8 Memcached 周边技术
 - 2.2.4.8.1 Twemcache
 - 2.2.4.8.2 Twemproxy
 - 2.2.4.8.3 Mcrouter
- 2.2.5 Tair
 - 2.2.5.1 Tair总体架构
 - 2.2.5.2 Config Server
 - 2.2.5.3 Data Server
 - 2.2.5.4 Tair高可用和负载均衡
 - 2.2.5.5 数据迁移
 - 2.2.5.6 存储引擎
 - 2.2.5.7 key/value/prefix

07 分布式缓存

模块 目标

- 01 熟悉Redis使用及原理
- 02 精通分布式锁并对其原理有深入理解

Redis原理与实战 分布式锁 其他常见缓存

- 2.2.6 Twitter开源缓存代理Twemproxy
- 2.2.7 EVCache
 - 2.2.7.1 EVCache使用场景
 - 2.2.7.2 EVCache的性能
 - 2.2.7.3 EVCache的高可用
- 2.2.8 Aerospike
 - 2.2.8.1 Aerospike具体实现
 - 2.2.8.2 Aerospike集群配置
 - 2.2.8.3 Aerospike与Redis的对比
 - 2.2.8.4 Aerospike在广告行业的具体应用
- 2.2.9 Redis
 - 2.2.9.1 Redis数据结构
 - 2.2.9.1.1 字符串 string
 - 2.2.9.1.2 列表 list
 - 2.2.9.1.3 字典 hash
 - 2.2.9.1.4 集合 set
 - 2.2.9.1.5 序列列表 zset
 - 2.2.9.2 Redis 的优点
 - 2.2.9.3 Redis 的安装
 - 2.2.9.4 持久化
 - 2.2.9.4.1 持久化发生了什么 | 从内存到磁盘
 - 2.2.9.4.2 如何尽可能保证持久化的安全
 - 2.2.9.4.3 AOF和RDB
 - 2.2.9.5 Redis 集群
 - 2.2.9.5.1 集群的作用
 - 2.2.9.5.2 集群体验
 - 2.2.9.5.3 主从复制

07 分布式缓存

模块 目标

- 01 熟悉Redis使用及原理
- 02 精通分布式锁并对其原理有深入理解

Redis原理与实战 分布式锁 其他常见缓存

- 2.2.9.5.4 哨兵模式
- 2.2.9.5.5 Cluster模式
- 2.2.9.5.6 Jedis操作集群
- 2.2.9.5.7 数据分区方案
 - 哈希值 % 节点数
 - 一致性哈希分区
 - 带有虚拟节点的一致性哈希分区
- 2.2.9.6 布隆过滤器
 - 2.2.9.6.1 布隆过滤器是什么
 - 2.2.9.6.2 布隆过滤器的使用场景
 - 2.2.9.6.3 布隆过滤器原理解析
- 2.2.9.7 深度原理
 - 2.2.9.7.1 删除策略
 - 2.2.9.7.2 io多路复用模型
- 2.2.9.8 企业级解决方案
 - 2.2.9.8.1 持久化
 - 2.2.9.8.2 事务
 - 2.2.9.8.3 分布式锁
 - 为何需要分布式锁
 - Java 中实现的常见方式
 - Redis 分布式锁的问题
 - 锁超时
 - 单点/多点问题
 - GC 可能引发的安全问题
 - Redis 分布式锁的实现
 - Redis缓存一致性

07 分布式缓存

模块 目标

- 01 熟悉Redis使用及原理
- 02 精通分布式锁并对其原理有深入理解

Redis原理与实战 分布式锁 其他常见缓存

2.2.9.8.4 缓存

缓存预热

缓存雪崩

缓存击穿

缓存穿透

多级缓存

性能指标监控

2.2.9.8.5 缓存协议

2.2.9.8.6 缓存连接池

2.2.9.8.7 管理缓存

2.2.9.8.8 缓存可用性

2.2.9.8.9 数据一致性

2.2.10 缓存系统基于Redis的设计与实践

08 并发编程

模块 目标

- 01 高性能落地
- 02 熟练使用线程池，了解其原理
- 03 对算法有较深理解

Java内存模型 AQS 线程池

- 2.3.1 Java内存模型
 - 2.3.1.1 Java的并发采用的是共享内存模型
 - 2.3.1.2 JVM对Java内存模型的实现
 - 2.3.1.3 硬件内存架构
 - 2.3.1.4 Java内存模型和硬件架构之间的桥接
 - 2.3.1.5 支撑Java内存模型的基础原理
- 2.3.2 线程之间的通信
- 2.3.3 线程之间的同步
- 2.3.4 线程安全性
- 2.3.5 原子操作类
 - 2.3.5.1 原子更新数组类型
 - 2.3.5.2 原子更新引用类型
 - 2.3.5.3 原子更新字段类型
- 2.3.6 上下文切换
- 2.3.7 synchronized的实现原理及应用
- 2.3.8 volatile原理及应用
- 2.3.9 线程不安全类
- 2.3.10 AQS
 - 2.3.10.1 AQS的设计和结构
 - 2.3.10.2 设计思想
 - 2.3.10.2.1 同步状态
 - 2.3.10.2.2 阻塞
 - 2.3.10.2.3 队列
 - 2.3.10.2.4 条件队列
 - 2.3.10.3 源码分析
- 2.3.11 CountdownLatch
 - 2.3.11.1 使用场景
 - 2.3.11.2 源码分析

08 并发编程

模块 目标

- 01 高性能落地
- 02 熟练使用线程池，了解其原理
- 03 对算法有较深理解

Java内存模型 AQS 线程池

- 2.3.12 Semaphore
 - 2.3.12.1 使用场景
 - 2.3.12.2 源码分析
- 2.3.13 CyclicBarrier
 - 2.3.13.1 使用场景
 - 2.3.13.2 源码分析
- 2.3.14 ReentrantLock
 - 2.3.14.1 可重入性原理
 - 2.3.14.2 获取和释放锁
 - 2.3.14.3 公平和非公平锁
- 2.3.15 Condition
- 2.3.16 线程池
 - 2.3.16.1 线程池核心原理
 - 2.3.16.2 ThreadPoolExecutor
 - 2.3.16.3 Executor框架
 - 2.3.16.4 线程池实战

09 数据结构与算法

模块 目标

01 面试必备

数据结构与算法

2.4.1 数组、链表、队列、栈

2.4.2 树、图、堆

2.4.3 排序、递归、回溯

2.4.4 深度与广度优先搜索

2.4.5 动态规划

10 微服务框架

模块 目标

- 01 精通微服务架构，分布式系统原理
- 02 熟悉主流的微服务框架
对服务治理有深入理解
- 03 熟悉微服务框架机制与原理
- 04 对RP框架C、序列化、服务治理有对应了解

分布式架构 Dubbo原理与实战

- 3.1 分布式架构演进
 - 3.1.1 单体
 - 3.1.2 垂直架构
 - 3.1.3 分布式
 - 3.1.4 微服务
- 3.2 分布式架构
 - 3.2.1 概念
 - 3.2.1.1 并发/并行
 - 3.2.1.2 线程
 - 3.2.1.3 锁
 - 3.2.1.4 集群
 - 3.2.1.5 重试与幂等
 - 3.2.1.6 无状态
 - 3.2.1.7 分布式锁
 - 3.2.1.8 分布式事务
 - 3.2.1.9 CAP
 - 3.2.1.10 Raft
 - 3.2.1.11 Lease
 - 3.2.1.12 "脑裂"问题
 - 3.2.1.13 Quorum NWR
 - 3.2.1.14 MVCC
 - 3.2.1.15 Gossip
- 3.3 分布式架构设计
 - 3.3.1 心跳检测
 - 3.3.2 负载
 - 3.3.3 全局ID
 - 3.3.4 幂等
 - 3.3.5 路由表

10 微服务框架

模块 目标

- 01 精通微服务架构，分布式系统原理
- 02 熟悉主流的微服务框架
对服务治理有深入理解
- 03 熟悉微服务框架机制与原理
- 04 对RP框架C、序列化、服务治理有对应了解

分布式架构 Dubbo原理与实战

- 3.3.6 分库分表
- 3.3.7 分布式存储
- 3.3.8 稳定性、可用性、可扩展性
- 3.4 分布式架构通信基础
 - 3.4.1 BIO、AIO、NIO
 - 3.4.2 Netty
 - 3.4.3 分布式通信框架
- 3.5 Zookeeper
 - 3.5.1 核心概念
 - 3.5.2 使用场景
 - 3.5.3 实战
- 3.6 Dubbo
 - 3.6.1 Dubbo介绍
 - 3.6.2 节点与节点角色
 - 3.6.3 服务暴露与服务消费过程
 - 3.6.4 服务调用过程
 - 3.6.5 api与spi
 - 3.6.6 服务消费方发送请求
 - 3.6.7 常用配置
 - 3.6.8 多协议支持
 - 3.6.9 do注册中心支持
 - 3.6.10 多版本支持
 - 3.6.11 异步调用
 - 3.6.12 主机绑定
 - 3.6.13 集群容错
 - 3.6.13.1 failover
 - 3.6.13.2 failfast
 - 3.6.13.3 failsafe

10 微服务框架

模块 目标

- 01 精通微服务架构，分布式系统原理
- 02 熟悉主流的微服务框架
对服务治理有深入理解
- 03 熟悉微服务框架机制与原理
- 04 对RP框架C、序列化、服务治理有对应了解

分布式架构 Dubbo原理与实战

- 3.6.13.4 failback
- 3.6.13.5 forking
- 3.6.13.6 broadcast
- 3.6.13.7 Cluster实现类分析
- 3.6.14 负载均衡
 - 3.6.14.1 Random
 - 3.6.14.2 RoundRobin
 - 3.6.14.3 LeastActive
 - 3.6.14.4 ConsistentHash
- 3.6.15 服务化最佳实践
 - 3.6.15.1 分包
 - 3.6.15.2 粒度
 - 3.6.15.3 版本
 - 3.6.15.4 兼容性
 - 3.6.15.5 枚举
 - 3.6.15.6 序列化
 - 3.6.15.7 异常
 - 3.6.15.8 调用
 - 3.6.15.9 常用配置

11 Spring Cloud

模块 目标

- 01 Spring Cloud 全家桶
- 02 spring cloud alibaba

Spring Cloud 全家桶 spring cloud alibaba

- 3.8 Spring Cloud
 - 3.8.1 Hystrix
 - 3.8.1.1 Hystrix处理流程
 - 3.8.1.2 容错
 - 3.8.1.2.1 资源隔离
 - 3.8.1.2.2 熔断
 - 3.8.1.2.3 回退降级
 - 3.8.1.3 降级策略
 - 3.8.1.3.1 自动开关降级
 - 3.8.1.3.2 人工开关降级
 - 3.8.1.3.3 读服务降级
 - 3.8.1.3.4 写服务降级
 - 3.8.2 Sentinel
 - 3.8.2.1 定义资源
 - 3.8.2.1.1 主流框架的默认适配
 - 3.8.2.1.2 抛出异常的方式定义资源
 - 3.8.2.1.3 返回布尔值方式定义资源
 - 3.8.2.1.4 注解方式定义资源
 - 3.8.2.1.5 步调用支持
 - 3.8.2.2 规则的种类
 - 3.8.2.2.1 流量控制规则 (FlowRule)
 - 3.8.2.2.2 熔断降级规则 (DegradeRule)
 - 3.8.2.2.3 系统保护规则 (SystemRule)

11 Spring Cloud

模块 目标

- 01 Spring Cloud 全家桶
- 02 spring cloud alibaba

Spring Cloud 全家桶 spring cloud alibaba

- 3.8.2.2.4 访问控制规则 (AuthorityRule)
- 3.8.2.2.5 热点规则 (ParamFlowRule)
- 3.8.2.2.6 查询更改规则
- 3.8.2.2.7 定制自己的持久化规则
- 3.8.2.2.8 规则生效的效果
- 3.8.3 Sentinel、Hystrix、resilience4j对比
- 3.8.4 网关
 - 3.8.4.1 API网关
 - 3.8.4.2 统一网关设计
 - 3.8.4.2.1 异步化请求
 - 3.8.4.2.2 链式处理
 - 3.8.4.2.3 业务隔离
 - 3.8.4.2.4 请求限流
 - 3.8.4.2.5 熔断降级
 - 3.8.4.2.6 泛化调用
 - 3.8.4.2.7 管理平台
 - 3.8.4.3 Zuul
 - 3.8.4.3.1 创建一个zuul-proxy模块
 - 3.8.4.3.2 启动相关服务
 - 3.8.4.3.3 配置路由规则
 - 3.8.4.3.4 过滤器
- 过滤器类型
- 过滤器的生命周期

11 Spring Cloud

模块 目标

01 Spring Cloud 全家桶

02 spring cloud alibaba

Spring Cloud 全家桶 spring cloud alibaba

自定义过滤器

核心过滤器

3.8.4.3.5 Ribbon和Hystrix的支持

3.8.4.4 Spring Cloud Gateway

3.8.4.4.1 核心概念

3.8.4.4.2 创建 api-gateway模块

3.8.4.4.3 Route Predicate 的使用

After Route Predicate

Before Route Predicate

Between Route Predicate

Cookie Route Predicate

Header Route Predicate

Host Route Predicate

Method Route Predicate

Path Route Predicate

Query Route Predicate

RemoteAddr Route Predicate

Weight Route Predicate

3.8.4.4.4 Route Filter 的使用

3.8.4.4.5 结合注册中心使用

使用动态路由

使用过滤器

3.8.4.4.6 Spring-Cloud-Gateway 源码解析

11 Spring Cloud

模块 目标

- 01 Spring Cloud 全家桶
- 02 spring cloud alibaba

Spring Cloud 全家桶 spring cloud alibaba

- 3.8.4.5 Spring Cloud Config
 - 3.8.4.5.1 核心概念
 - 3.8.4.5.2 在Git仓库中准备配置信息
 - 3.8.4.5.3 创建config-server模块
 - 3.8.4.5.4 通过config-server获取配置信息
 - 3.8.4.5.5 创建config-client模块
 - 3.8.4.5.6 配置中心添加安全认证
- 3.8.4.6 Spring Cloud Bus
 - 3.8.4.6.1 动态刷新配置
 - 3.8.4.6.2 配合WebHooks使用
 - 3.8.4.6.3 给config-server/server添加消息总线支持
- 3.8.4.7 Spring Cloud Stream
 - 3.8.4.7.1 快速实战
 - 3.8.4.7.2 抽象绑定器 (The Binder Abstraction)
 - 3.8.4.7.3 发布-订阅(Persistent Publish-Subscribe Support)
 - 3.8.4.7.4 消费组 (Consumer Groups)
 - 3.8.4.7.5 消费者类型 (Consumer Types)
 - 3.8.4.7.6 分区支持 (Partitioning Support)
 - 3.8.4.7.7 编程模型 (Programming Model)
 - 3.8.4.7.8 自定义消息通道 发送与接收
 - 3.8.4.7.9 消费组和分区的设置
- 3.8.4.8 服务发现

11 Spring Cloud

模块 目标

01 Spring Cloud 全家桶

02 spring cloud alibaba

Spring Cloud 全家桶 spring cloud alibaba

3.8.4.8.1 客户端服务发现

3.8.4.8.2 服务端服务发现

3.8.4.8.3 服务注册中心

3.8.4.8.4 Self-Registration

3.8.4.8.5 Third-Party Registration

3.8.4.8.6 常见的服务发现框架

ZooKeeper

Etc

Consul

Nacos

Eureka

Eureka核心概念

Eureka 工作流程

给Eureka注册中心添加认证

源码分析

3.8.4.9 Spring Cloud Sleuth

3.8.4.9.1 给服务添加请求链路跟踪

3.8.4.9.2 整合Zipkin获取及分析日志

3.8.4.9.3 使用Elasticsearch存储跟踪信息

3.8.4.9.4 Logstash

集中、转换和存储数据

整合logstash

11 Spring Cloud

模块 目标

- 01 Spring Cloud 全家桶
- 02 spring cloud alibaba

Spring Cloud 全家桶 spring cloud alibaba

安装logstash

3.8.4.9.5 跟踪原理

3.8.4.9.6 收集原理

3.8.4.9.7 拓展视野

淘宝“鹰眼”

京东的“Hydra”

大众点评的“CAT”

微博Watchman

12 大数据

模块 目标

- 01 对大数据有基础了解
- 02 熟悉流批一体，提升竞争力

ES

ES

3.7.1 索引、文档、节点、集群、分片、副本

3.7.2 聚合分析

3.7.3 多字段特性

3.7.4 倒排索引

Flink

Flink

dataStream

watermark

TimeWindow

SessionWindow

CountWindow

13 云原生

模块 目标

- 01 从事过大规模分布式系统资源管理和任务调度运维经验，熟悉一些k8s Yarn mesos等开源技术
- 02 熟悉大规模集群自动化运维系统开发以及集群故障管理、故障容错和自愈
- 03 熟悉整个devops体系有过从0到1，落地的经验的优先

Dockcer、K8s、Istio

- 3.8.4.10.1 Docker
- 3.8.4.10.2 K8s
- 3.8.4.10.3 Cloud Foundry
- 3.8.4.10.4 Mesos+Marathon
- 3.8.4.10.5 容器弹性扩缩容
- 3.8.4.10.6 DevOps
- 3.8.4.10.7 持续集成
- 3.8.4.10.8 代码质量管理
- 3.8.4.10.9 性能监控
- 3.8.4.10.10 Istio使用
- 3.8.4.10.11 Istio流量管理
- 3.8.4.10.12 Istio实战

14 实战

模块 目标

- 01 宏观把握项目整体架构
- 02 老师倾囊相授多年的工具包

大厂真实技术架构解析 珍藏的工具包、脚手架

大厂项目必备工具、开源包等使用详解
淘宝架构演进
亿级流量直播技术架构
微博feed流设计架构

14 实战

模块 目标

- 01 学习大厂真正的秒杀
- 02 百万级qps的解决方案

秒杀系统实战

4.3 秒杀系统实战

4.3.1 项目流程图、涉及知识点

4.3.2 表结构设计

4.3.2.1 商品表

4.3.2.2 商品库存表

4.3.2.3 活动信息表

4.3.2.4 订单表

4.3.2.5 用户信息表

4.3.3 持久层架构搭建

4.3.3.1 引入MyBatis依赖

4.3.3.2 反向代码生成器

4.3.4 库存模块

4.3.5 订单模块

4.3.5.1 订单号生成 SnowFlake

4.3.5.2 下单核心逻辑

4.3.6 超时未支付订单处理逻辑

4.3.6.1 Dead Letter Queue

4.3.6.2 补偿机制

4.3.7 商品模块

4.3.7.1 查询产品列表

4.3.7.2 查询产品详情

4.3.8 秒杀活动模块

4.3.9 用户模块

4.3.10 消息通知模块

4.3.11 热key探查

4.3.11 限流模块

4.3.12 邮件模块

4.3.13 压测

14 实战

模块 目标

01 hc最多的金融平台

聚合支付平台

4.4 聚合支付平台 去Spring化 从0-1架构详解

4.4.1 支付业务基础知识、支付原理、支付接口常见设计方案

4.4.2 平台功能、架构解析、技术栈介绍

4.4.3 核心功能：基础框架详解 下游渠道选择路由器详解 支付网关对接详解

模块 目标

01 复杂权限系统设计方案

ACS

4.5 ACS

4.5.1 功能权限模型实现

4.5.2 功能filter过滤器实现

4.5.3 业务系统集成功能权限验证

4.5.4 缓存设计方案

模块 目标

01 提现技术硬实力的项目

手写基础正则表达式解析器

4.7 手写基础正则表达式解析器

正则表达式基础教程

正则表达式转化为NFA

NFA到DFA的转化

DFA优化

Vert.x 异步编程实战

Vert.x 异步编程实战

reactive编程模式介绍

Vert.x 框架介绍

Vert.x web application实战

模块 目标

01 出行项目

出行项目

出行项目-业务整体纵览

数据平台的构建思路

出行项目的业务内容

Spark、Flink、Scala基础知识

出行项目-大数据环境准备

大数据平台的搭建

Flink、Spark的技术比较

大数据环境下，如何进行数据同步？

出行项目-数据的落地

Spark作业监控、维护

Kafka、Flink流式应用的 exactly once 语义

车辆轨迹实时回放业务实现

出行项目-核心问题解决

Kafka消费者准确性问题

Hbase介绍、数据倾斜问题处理

SparkSQL开发

出行项目-监控调优

SparkStreaming 限流、背压、冷启动

Spark任务监控

15 面试

模块
目标

01 面试辅导

面试辅导

5 软实力-求职面试深度辅导

5.1 简历

5.1.1 简历模板分享

5.1.2 简历书写注意点

5.1.3 大厂内推

5.2 面试

5.2.1 一面：基础能力，重广度

5.2.2 二面：项目能力，重深度

5.2.3 三面：行业领域经验

5.2.4 四面：HR软素质考察

5.3 工作经验分享

5.4 面试经验分享

往期学员offer展示

入职公司	个人信息	薪资涨幅
 百度	谭同学 (Java技术专家)	167% ↑
 蚂蚁金服 ANT FINANCIAL	魏同学 (java技术专家)	88% ↑
 唯品会	赵同学 (java架构师)	96% ↑
 美团	何同学 (高级java工程师)	101% ↑
 得物	徐同学 (java研发工程师)	50% ↑
 Alibaba Group 阿里巴巴集团	马同学 (java高级工程师)	61% ↑
 跟谁学	高同学 (java开发工程师)	67% ↑
 58同城	马同学 (资深java架构师)	74% ↑
 金商	黄同学 (java技术工程师)	53% ↑
 滴滴	彭同学 (java框架研发工程师)	78% ↑
 美团	程同学 (资深java开发工程师)	84% ↑
 Tencent 腾讯	刘同学 (高级java开发工程师)	93% ↑

更多offer请联系我们

部分面试官直推企业





爪哇教育

UID: 526153452