

用Go实现一个仿小红书发帖功能 Part3

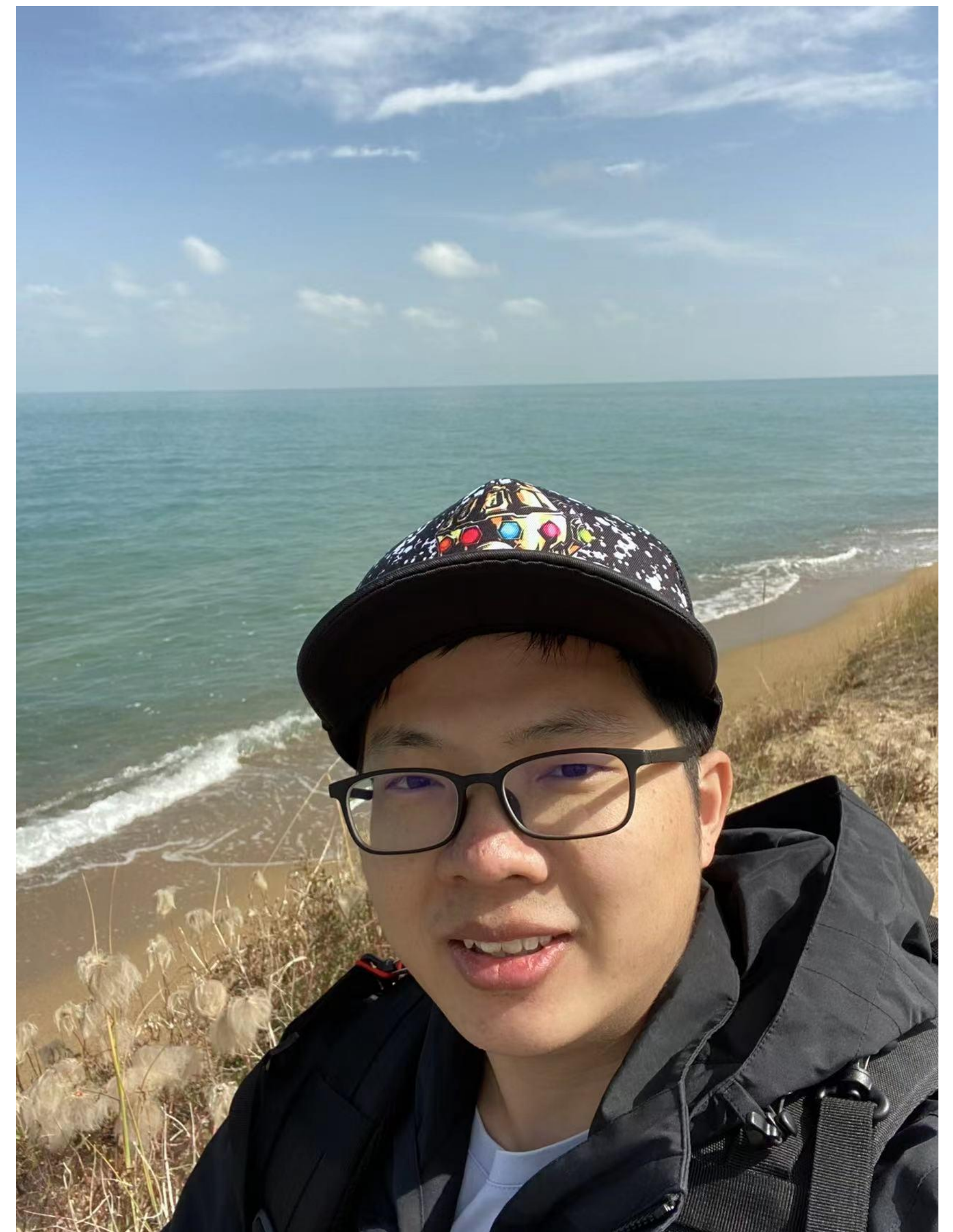
大明

2023.05



自我介绍

- 咸鱼一条，还是比较咸的那条
- 工作唯唯诺诺，开源重拳出击
- Java Go 两开花
- 业务开发、中间研发一把抓



目录

- 为什么学习 Go
- 如何学习 Go
- 如何成为前 20 % 的程序员
- 怎么成为公司卷王
- 怎么成为 offer 收割机

为什么学习 Go?

- 云原生还在持续不断地改造行业，即便是最顽固的传统行业，也开始拥抱云原生。而最适合云原生的就是 Go 语言。
- Go 在侵占部分 PHP 和 Python 的岗位。我观察到的趋势就是现在比较多新起来的业务都是用 Go 了。而且，PHP 或者 Python 在系统成长到一定地步之后，如果考虑重构，那么 Go 也是一个首选语言。
- Go 相比 Java，资源消耗更少。中小型公司买服务器也不需要那么多，所以采用 Go 作为第一语言的公司也越来越多。
- Go 的生态还不完善。这算是利弊都有。弊端就是一些高端业务，比如说金融类的基本上还是用 Java，但是好处就是没那么卷，你还有发挥空间，尤其是开源的发挥空间。

我一向鼓励大家多学几门语言。也就是说 PHP、Java、Python 什么的不是互斥的，你可以都学学。而且，现在的业界趋势就是异构系统，一个系统不同部分用不同语言，因此掌握多门语言是必要的。

如何学习 Go：基本步骤

分成几个步骤：

- **先掌握 Go 的基本语法**。这个过程你可以考虑用 Go 来造一些工具类，比如说用 Go 泛型造各种数据类型，包括链表、红黑树、跳表之类的。等你开始学并发编程的时候，就造一些并发数据类型，比如说照着 Java 的并发集合，用 Go 抄一份过来。
- **掌握 Gin、GORM、gRPC 等框架的用法**。因为这些框架主要是用来开发业务代码的，所以你可以尝试自己用它们来造一个小型的 Web 应用，就当做是练习。
- **为 Gin、GORM、gRPC 等开源提供扩展插件**。你可以找到你喜欢的 Go 开源框架，然后给它们提供一些开源框架。比如说你完全可以为 Gin 提供一整套的插件，从解决跨域问题到解决限流问题.....
- 如果自己是跨语言的，那么看看自己使用的语言里面有什么好用的东西，但是 Go 没有的，那你就**造一个轮子出来**。

到这个地步你基本上 Go 语言就很擅长了。大部分人学习新语言的困境就是练习太少，缺乏使用场景。所以看了语法又忘了。

如何学习 Go：应付面试

应付面试就要讲究一个深挖了，你需要准备好以下内容。

- Go 内置类型的实现原理，包括切片、map、channel。
- Go sync 包各种数据结构以及底层原理，尤其是锁的实现原理。
- GMP 调度，纯八股文。
- Go 内存分配和垃圾回收，纯八股文。

如果你面的不仅仅是业务开发，还倾向于中间件研发，那么这两个你就要投入更加多的时间：

- Go 并发编程，要能应付面试官现场出题。
- Go 网络编程，接近八股文，要能利用 Go 的 net 包设计一点简单的网络协议。

如何成为前 20% 的程序员 —— 打铁还得自身硬 极客时间 | 训练营

- **专**：某个技术领域深挖。比如说云原生开发，或者说网关、数据库分库分表、大规模分布式系统、高可用微服务架构.....你首先要成为某个领域的专家。这是一个日积月累的事情。
- **广**：在你专研的领域之外，保持学习精神，做到对相关行业有一个基本的了解。

你需要在行业里面有一定的影响力：

- 参加各种线上、线下 meetup，最好是作为嘉宾参加，而不是作为观众参加。
- 去开源，成为知名开源框架的核心成员、负责人。
- 输出标准、实践：如果不能输出，就参与进去。

如何成为前 20% 的程序员 —— 打铁还得自身硬 极客时间 | 训练营

- **专：某个技术领域深挖。**比如说云原生开发，或者说网关、数据库分库分表、大规模分布式系统、高可用微服务架构.....你首先要成为某个领域的专家。这是一个日积月累的事情，相信一万个小时定律。
 - **熟知该领域内的各种解决方案**，并且尝试在实践中落地，总结优缺点。
 - **精通该领域内的某几个主流框架**，要做到原理了如指掌，二次开发毫无困难。
 - **在该领域内要有一个拿得出手的作品。**如果公司欠缺机会，就去开源。
- **广：在你专研的领域之外，保持学习精神，做到对相关行业有一个基本的了解。**
 - 确保虽然没吃过猪肉，但是至少见过猪跑
 - 保持对技术前沿的关注、了解当下趋势

如何成为前 20% 的程序员 —— 影响力

在公司有影响力：

- 当他们讨论到某一个问题的時候，就会想到你
- 遇到疑难杂症都会来找你
- 持续在组内输出规范、最佳实践

在行业里面有一定的影响力：

- 参加各种线上、线下 meetup。最好是作为嘉宾参加，而不是作为观众参加。
- 去开源，成为知名开源框架的核心成员、负责人。
- 输出标准、实践：如果不能输出，就参与进去。

影响力的核心就是：你要学会装逼。用作品证明自己，用嘴炮说服别人。

全是个人经验，没有丝毫客观性的建议：

- 评价一件事能不能做，要不要做得很好的唯一标准就是：这个东西能不能让你升职加薪？如果这个标准还不够，那就加一条：这个东西能不能让你的领导升职加薪？
- 保持高效率，偶尔比领导认为的快一点。既不要太频繁，也不要快太多，不然领导就知道自己傻逼了，估计你的能力不足。剩下的时间，用来提升自己。
- 在能决定你晋升的人面前，做一条舔狗。你直属领导一般只有建议权，你 +2 的那个才有决定权。抓住机会就使劲表现。要日积月累，水滴石穿。
- 任何领导认为是你应该做的，达标就好，因为做得再好，“都是你应该做的”。那些本来不是你应该做到的，你才能做到“超出预期”。
- 不间断在组内扩大自己的影响力。各种方案评审、技术相关复盘、组内技术讨论，言之有物。遇到系统出了奇怪的 BUG，主动参与 DEBUG。要注意凸显：自己见识广博、积累深厚。
- 成为救火专家。如果一个东西可能出问题，但是又不是你负责的，就让它出问题，而后你再以救世主的形象降临救火。
- 和同事搞好关系，把朋友搞得多多的，把敌人搞得少少的。

怎么成为 offer 收割机

长期积累：

- 入职的第一天，就要想好自己下一跳的话，**要在这家公司积累什么项目经验。**
- **利用开源来补足短板。**如果你的公司不能提供很好的项目经验，就去开源里面找。

短期冲刺：

- 像背政治历史那样背八股文。
- 仔细梳理自己的项目经验，要综合考虑面试官可能问哪些问题，怎么引导话题。一些关键问题，要提前写好答案，然后背下来。
- 针对不同的公司和岗位调整简历，调整面试方向、话术。
- 努力训练营毕业，找我做简历咨询。

求职过程：

- **脸皮要厚：**不要看什么岗位描述，反正看到公司合适，薪资水平合适你就投简历，大不了就是没下文。
- **多练习：**多面几家，而后你就知道当下的市场需求，面试的方向，你简历的薄弱点。**根据面试反馈调整简历和面试策略。**

Q&A