



下载APP

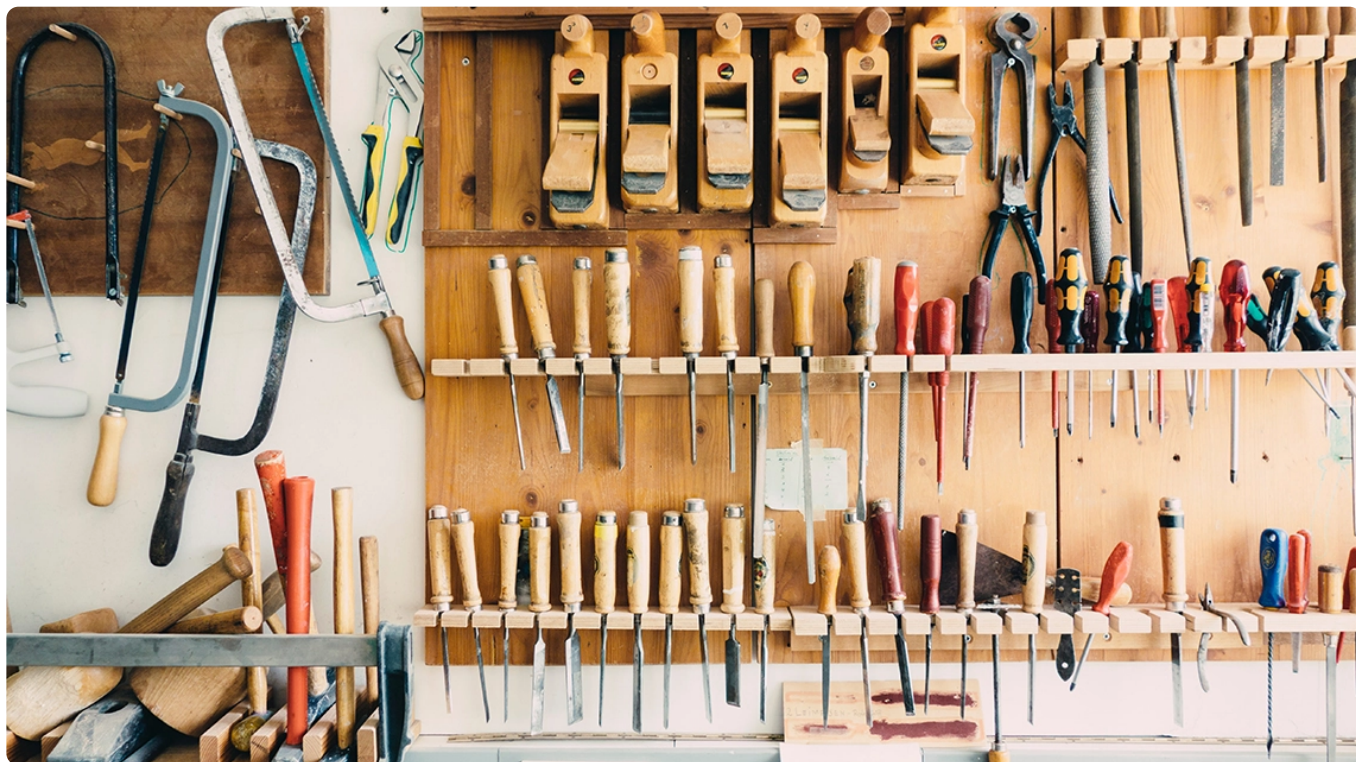


大咖助阵 | 飞雪无情：十年面试经验忠告，不要被框架所束缚

2021-11-26 飞雪无情

《手把手带你写一个Web框架》

课程介绍 >



讲述：正霖

时长 11:27 大小 10.50M



你好，我是飞雪无情，在 IT 互联网行业已经工作十多年了，目前是公司的产品研发负责人，带领 100 多人的团队。

作为一位 Go 语言的布道者，我在自己的博客里，写过 [《Go 语言实战笔记》](#) 系列，共计 30 篇 15 万字，也写过 [《Golang Gin 实战》](#) 系列，共 14 篇十多万字；也是公众号飞雪无情的主理人，写过不少关于 Go 和 Web 框架的文章，积累了小几万粉丝。

写了这么多，但当我收到极客时间的约稿，是不太敢接的，因为实在不知道写什么你的收获更大。



如果写框架本身吧，不管是这个专栏，还是我自己的公众号都已经写了很多了，再写就是老生常谈了，你也不一定会感兴趣。如果写管理吧，虽然是个好的话题，结合我的从业经

验也能分享一些心得，但是和专栏的主题有点格格不入。

我仔细思考了很久，正好最近因为频繁招聘，对面试和工作表现差异的感受比较强烈，最终选定了今天这个话题，想结合我自身学习、面试的经验，给你分享一下如何更好地学一门框架、如何不让自己成为框架的附庸，不至于在框架更新换代时落伍，失去了竞争力。

误于面试

去年我离开了工作十年的老东家金蝶随手记，来到一家不算太大，但是很有朝气、提倡创新的新公司十方教育集团，负责整个产品研发的工作。新的岗位对我来说是有很大的挑战的，不管是团队的搭建、流程的建立，还是规范的梳理，都需要投入大精力去做。

这其中最具有挑战的就是团队的扩充以及梯队的搭建。因为公司业务发展，产品研发团队要大量扩充，所以我今年 3、4 月份的工作大部分都是招聘、面试。

在面试的过程中我发现，不少面试者讲起来基础还是可以的，问一些基本原理也都能讲明白，一些常用的框架也都用的比较熟。但是，**如果你问他以前做的项目，就会发现很一般，也很少用到他所说的这些面试知识。**当你给他出个题目，想看看他的实战经验，**也看不到 3 年、5 年经验应有的实战能力。**

比如很多面试者都可以使用 Gin 框架快速开发一个 API、写一个中间件，但是如果要问 Gin 如何快速匹配到一个路由的？为什么可以这么快？中间件是如何拦截请求的？转交给下一个中间件是如何做到的？能答上来的就很少了，更不用说再细问一步，中间件是使用的哪个设计模式？自己如何实现一个？

这种虽然能回答一些原理，但深入再问，能答上来的寥寥无几的经典面试情况，就是因为**很多人的原理，是从面经上看的、背下来的，没有自己动手去验证过。**加上市面上有很多现成的框架帮助他们完成日常工作，止步于会用就行，导致了解的知识都是皮毛，经不起考验。

为了面试而学并不是太新鲜的事情，十多年前也有，但是没有现在那么厉害，最近几年因为就业压力导致刷题盛行，我们只能被逼着去学，不然面试第一关都过不了。

但是面试不只一关，所经历的公司也不止这一家，止步面试的学习也会明显跟不上技术的迭代。因为就算你面试过了，还有试用期，进入工作后，自己的不足就会暴露出来，甚至

出现更严重的问题，影响你的转正。

止于工具

现在技术更新是日新月异的，框架更是如此，隔一段时间换一个，不同的公司很大可能采用不同的框架，所以你换新工作之后，一般都得重新学习一门新框架，这会让你以前的经验打折扣，如果新公司用的框架你不会，自己的竞争力会大大降低，谈薪水都没底气。

但是你会发现，有的人能迅速上手新框架，但是有的人接受速度很慢，逐渐就被淘汰了。

出现这种情况有四层原因：

1. 进取心不足，没有学习其他框架。
2. 有进取心，但是太相信自己在用的框架，投入太多，甚至把它当成自己的砝码，没有精力学习其他的，也不想学习。
3. 有关于框架发展的想法，但是关注度不够，没有跟上趋势。
4. 有关关注也跟上趋势了，但是基础不牢，没有真正理解框架的底层原理，进而学习其他框架比较慢。

表面看是四层原因，其实深究都是被框架给“拖累”了，没有把主动权抓在自己的手里。

框架只是一个工具而已，是我们实现自己价值的多种工具中的一种。不止框架是工具，我们的编程语言也可以是工具。所以框架可以一个个的换，编程语言也可以，现在很多服务已经使用 Go 语言替代 Java、Python 了。

就像每一种编程语言都有它的使用场景，每一种框架都有它的优势。如果旧框架被替代，那么说明新框架有旧框架无法满足的优势，**我们要做的就是根据需求，选择更适合的框架，最大程度提升自己的核心竞争力。**

所以，请把框架当工具，不要让它成为你的负担，框架被淘汰，我们不能跟着一起被淘汰。

既然为面试而学只能解决一时问题，进入公司后无法更快适应公司技术栈，更快地做出成绩，那么日常怎样学习才能更好地提升自己，让自己越来越具备竞争力，甚至被各大公司

抢着要呢？

始于基础

基础得学牢固这一点我是有切身体会的。这里说的基础指的是，**编程语言本身的功能、概念、语法、模式等，它是编程的地基，不会随着框架的改变而改变。**

大学时，在学会使用 SSH（10 年前的 J2EE 三大经典框架 Spring、Hibernate、Struts）框架写出来一个管理系统，那会我感觉自己很牛，其他同学都不会，就我会。后来认识了一位已经工作的学长，他和我讲，好好把语言基础打牢，这些框架自然就会了。当时真的是很不理解这句话。

毕业后因为工作需要，用的框架换来换去，我才深刻理解到这句话的价值。大学我学的是 Struts1，毕业后就升级到了 Struts2，语法完全不一样，相当于得重新开始学一遍，再过一段时间，又被 SpringMVC 后来居上了，又再学一遍。现在的新同学估计都没有听过这几个框架。

几次三番换框架，就算我有进取心想紧跟趋势，也多少有点忙不过来了。又得回头重新巩固对 Java 基础的理解。

后来我换到 Go 语言也一样，比如 Go Web 框架，有 Martini、Beego、Iris、Echo，还有现在流行的 Gin 以及新兴的 Fiber，也是轮转多换，但是对于我们开发者来说，不变的是基础。

比如 Go 语言本身的基础，学会了这个，想学其他语言很容易；再比如 Go 语言 HTTP 处理、路由的基础，这是一切 Go 语言 Web 框架的前提，研究透了它们，不管你用什么 Web 框架，都可以庖丁解牛，搞得清清楚楚。

框架会经常变，但是不变的是这些基础。

精于原理

基础掌握好了，在学习具体框架的过程中，我们也容易走入另一个误区，那就是止步在使用上，没有把原理搞清楚，面试讲不出内部的实现逻辑、为什么这么实现，工作中面对新框架，也无法快速地熟悉使用，甚至二次开发。

所以当我们学习一门框架的时候，不仅要熟练地使用它，还要研究它里面的**功能是如何实现的、为什么这么做、如果自己来做，有没有更好的办法**？这样才能更好地使用。同时，在同类新框架出现的时候，也可以很快跟上，因为同一类框架，底层原理都是差不多的，就是一些实现方式、用户体验（开发者是用户）、周边生态等不太一样。

比如刚才说的那几个 Go 语言的 Web 框架，基本上都是基于 Go HTTP 的封装，让我们开发者可以更好地使用，但是在具体的实现上有所不同而已。如果你基于一个 Web 框架，把原理搞清楚，那再学习其他 Web 框架就会非常快，甚至自己撸一个 Web 框架都没问题。

对于一个框架的深入学习，**除了内部的实现原理，还要掌握一些模式的应用**。比如 MVC 模式、设计模式、算法等比较基础的能力，这些是构建一个可扩展、易用、高性能框架的基础；再比如常见的工厂模式、责任链模式等，会被大量用到框架的架构设计中。所以我們也要掌握这些。

用于实战

实践是检验真理的唯一标准，这句话说的真好，对于我们来说也一样。你说自己掌握了基础、精通了原理，那怎么衡量呢？就是靠实战。这是对结果的衡量，也是能体现自身价值、升职加薪的唯一方式。

在面试的时候，除了以前做的项目之外，我还比较喜欢问一个问题：**作为 5 年工作经验的你，为什么认为自己比其他 3 年工作经验的面试者要强？**

这个问题考察的点比较多，比如学习能力、总结能力、协调能力、经验等等，但是最终考察的其实是面试者解决问题的能力。能解决问题，才能证明你学的基础、说的原理、把框架当工具的认知有价值了，而不是纸上谈兵，更通俗点就是有用。

所以实战，才是我们“始于基础”、“精于原理”的最终目的，靠实战去实现自我的价值。那么怎样才可以提升自己的实战经验呢？这里我给出一些方法供你参考：

1. 积极参与公司项目，不要只为了完成任务，还要搞懂为什么这么做，背后的技术是什么；遇到问题时，也一定要查到根本原因，这些都可以提升你的实战经验。
2. 跟着高手学，比如一个项目立项、一个需求评审，都会有资深的开发甚至架构师牵头设计方案，要多跟着他们学，不懂的多问，尝试自己去设计，来锻炼自己的实战能力。

3. 如果工作中没有以上机会怎么办呢？自己去找开源项目，先熟悉它、使用它，然后参与贡献代码，这也是个好方法。
4. 现在很多技术网站都有问答，你可以去这个频道去回答别人的问题，通过这种输出倒逼自己输入的方式，也能提升自己的实战经验。

其实方法有很多，也不限于以上四种。你只要自己动手去做就对了。


总结

框架是减少重复代码、提升开发效率的工具，我们可以通过它打开学习框架的大门，研究它背后的原理，理解它的架构设计，进而提升我们自身的能力，但是记住，永远不要被它所束缚。

欢迎在留言区分享你的思考。如果你觉得有收获，也欢迎把今天的内容分享给你身边的朋友。我们下节课见。

分享给需要的人，Ta订阅后你可得 **20 元现金奖励**

 生成海报并分享

 赞 2  提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 大咖助阵 | 大明：抽象，抽象，还是抽象

下一篇 30 | 设计先于实战：需求设计和框架搭建

训练营推荐

Java 学习包免费领 NEW

面试题答案均由大厂工程师整理

阿里、美团等
大厂真题

18 大知识点
专项练习

大厂面试
流程解析

可复用的
面试方法

面试前
要做的准备

精选留言 (1)

写留言



Vincent Lv5

2021-11-30

始于基础，精于原理 + 实践检验真理

展开 ∨

