

49 临门一脚：前端面试的本质是什么？

更新时间：2020-06-19 10:47:03



“ 富贵必从勤苦得。——杜甫 ”

在专栏的最后，和大家聊聊我所理解的面试、以及进阶大厂的一些注意事项。通过这两节的沟通，我希望能传达给大家一个非常重要的认知：

面试的过程，其实是面试官在评估”是否愿意与你成为工作伙伴“的过程。

摆正心态

基于此，大家首先要搞清楚这场对话的一个大前提：它确实是一次考核，但并非一场考试。面试官不是高高在上，坐在讲台上喝茶看报的监考老师，而是与你站在平等地位的一个同行。

对面试官来说，重要的不是这道题你到底有没有 100%地答对（何况很多问题根本没有标准答案），而是你答题的过程中所折射出来的处理问题的态度和思考问题的方式——这些东西能够间接地说明你是一个什么样的人。

大厂的前端团队，不缺熟读框架原理和 `api` 用法的施工队员，更多是需要一个具备较强的抽象思维能力、能够 **cover** 各种问题和场景、进而独当一面的工程师。（大家先牢记这个观念，这是本文的重要基调，可以说后面我们许多具体问题的描述都离不开这一点）。

前端面试的评价逻辑

当面试官拒绝你时，他最有可能在想什么？

面试官的回绝，往往不是因为你在对答过程中的某一个点出了问题/某一个题没答上来，而是因为你的综合实力没能达到预期。

具体到面试过程中，你会发现，很少会有面试官对哪个具体的问题非常执着。甚至做算法题，你代码写得不完整、没办法 **cover** 所有的用例，这些都不是大问题，面试官一般会更关注你的思考过程、看你是否掌握了这一类算法的基本思路；没有用过某个轮子，也根本不是问题的关键，关键是你已经在用的那个轮子，你对它了解多少，基于这个轮子，你对框架的设计模式、架构思想又挖的有多深；至于学历，上文已经说过，这玩意儿对你的钳制非常有限。社招能拿到一面，说明面试官已经不 **care** 你是大专还是清华了，此刻他关注的是你的项目和能力。

综上所述，你与大厂之间，差的未必是那几个轮子——背 **Api**、背热门算法题这种事情，体现不了任何问题。失败过后，需要反思的不是“我还缺了哪个轮子、哪个工具”，而应该是“我还缺了哪些能力”。基于此，不妨试试问自己以下问题：

- 我是否能够独当一面？在面试过程中，我对自己经手的业务是否能充分理解其业务背景、选型依据和优化/重构过程？最重要的是，我能否把这一系列的东西说清楚？
- 我的特色是什么？技术攻关能力？业务 **Sense**？还是说别的什么？我是否有具体的成果来佐证我的特色？
- 我的技术基础是否存在漏洞？这一块，可以参照本文的“硬实力”板块，不必对每个点都熟稔于心，但若是存在一整块的知识盲区，那么情况就不太妙了。

以这些问题为线索，反思自己的面试过程，从能力的维度去做弥补和突破。而不要单纯地出于焦虑去学使用轮子的 **100 种姿势**。

至于各种培训广告里常见的“你离 **xx** 厂前端，只差一个 **xx** 框架！”之类的 **slogan**，大家还是忘了吧 **orz**。

不要迷信“面试宝典”

笔者的个人邮箱（xycoding@163.com）和微信号（[xyalinode](#)）都是开放的，因此平时有幸有触及到海量读者、听到各种各样的声音。这其中与面试有关的，一个非常有趣的问题是：“我买了那么多面试课，为啥还是找不到工作？”

这里就引出了另一个问题：“面经”、“指南”、“**xx** 大课”，海量参考信息，人人都可触及，却并非人人都能够成为出色的工程师——为什么？

因为用的方法不一样。

平时的工作里，定期更新自己的知识结构、留心观察比自己强的人写代码的手法。面试前，海量收集搜索引擎提供给自己的面经/真题，寻找出这些知识间的规律，建立一个基本的知识库。然后在海投面试的过程中，在实战的过程中去更新这些知识库。在这个过程中，把所谓“面试宝典”作为建立自己的知识结构的许多手段的一种。

是不是比面试就在眼前时，把自己的未来和命运一股脑全压在一本“面试宝典”上，要稳多了？

重要的是，你能否以适合自己的形式，去建立自己的知识结构，而不是到底刷了几篇文章几百道题。

“软技能”在面试中的实操办法

有同学一看到“软技能”就本能地会认为是“教你做人”系列，觉得这玩意儿太玄，太抽象，没必要看、没必要学，完全看命。仁者见仁，智者见智，在笔者看来，这东西非常“必要”。

这里所说的“软技能”，一点也不抽象，它非常具体，它指的就是你做成一件事的能力。

这里的“做成”往往不是你写一坨代码、或者造一个轮子这么简单。我自己是创业团队出身，我比各位都清楚，一个程序员在小团队待久了，容易形成怎样的思维定式——简单来说，“技术就是一切，技术强就完了，技术强就最屌，技术强就解决了 90%的问题”。这也是我在许多小团队“头羊”身上所看到的，即“完全 focus 于技术，技术外的思考和认知为 0”这种情况，每每遇此，都非常惋惜。

因为在大厂，单说技术，大家都很强。这种情况下，除非你的技术实力，是有着戈壁滩的十层沙都盖不住的那样的光芒，否则，你必须具备一些必要的软实力，才能在这里生存、呼吸——这种生存、呼吸的能力，就是面试官的重要关注点。

在面试过程中，软技能的凸显，基本集中在你对项目的阐述和理解环节，以及面试官刻意抛出的“闲聊时间”里。必要的面试“软技能”，及其对应的修炼方法，我这里总结如下：

1. 好好说话的能力

一个建议，一个观点，能不能在有限的时间内、用清晰有条理的语言把它表达出来？

不要小看“说话”这件事情。

举个反例，去年的事情，隔壁组的同事内推了一个候选人，一面二面对答如流，技术基础、技术产出都扎实得让人五体投地。然后总监面的时候，他挂了，二面面试官一脸懵逼跑去找老板问原因，答曰：“思路混乱，陈述啰嗦”，简单讲，就是话没说明白，沟通过程中给人的阻力比较大。挂在这点上，完全就应了我们开头说的那点——面试官找的是一个工作伙伴，沟通都困难，还咋做伙伴？

再举个正面例子。

高中同学，毕业创业失败，跑去找工作。创业期间他是 TL，而且是那种不关注技术细节、只做方案和 ppt 的 TL。我一直非常讨厌这种拿 ppt 作生产力的骚操作，但不得不说丰富的 presentation 经历使得这位老哥具备了我等普通开发所不具备的表达能力。最后这位老哥面试的时候，虽然面的是一水的一线开发岗，竟然没一个挂的。问及窍门，答曰“聊就完了，题能不能做对是另一说。把面试官聊开心了，就稳了”。最后去了头条，唔，在下佩服...

当然，并非每个人都具备如此清奇的骨骼和体质。但这也足够启发我们去认识到表达能力的重要性。因此，对于写进简历的项目，不能止步于“写上了”，然后全等着临场发挥（除非你对自己的即时表达能力有充分的自信）。而应该在面试前就整理出每一个项目对应的口头文稿，尝试先把它讲给你身边的人，同事或者同行业的好友，征询他人的意见，进而不断完善项目文稿。

2. 尊重他人的能力

这步很简单，我就说两点：

- 尊重“前任”。问及“为何从前司离职？”，可以说寻求更广阔的平台，可以说更看好贵司的业务，思路是使劲儿夸目标团队的好，万万不可对前团队恶言相向，表达出“因为讨厌之前的团队，故想来现在的团队”这样的信号。
- 尊重对话的对象。与面试官意见相左，不要吵架，不要吵架，不要吵架。即便你是正确的一方，也要学会尊重和引导自己的对手。因为候选人态度激烈进而觉得“有个性”决定收入麾下，只是极少数同样有个性的面试官会

做的事情，极少数极少数。

反例：年初，本市另一厂裁员过来的一位老哥，二面时面试官只是出于了解随口问起前团队的经历，这位老哥立刻开始大放厥词，从大 **leader** 喷到小 **leader** 喷到同桌。二面面试官挂他时毫不犹豫——现团队有一天可能也会变成前团队，不尊重前团队的人，很可能有一天也会对现团队破口大骂——不管你是不是真会这样做，在面试官的判断里，你大概率就是这样的一个人。

3. Owner 意识

前面我们说过，面试官找人，本质上是在给自己找工作伙伴。作为潜在的他的团队的一员，他在意的是：

- 是否能与你顺利的沟通（参见 1、2）
- 你能不能干活
- 你能不能扛事儿

其中“干活”说的是你把分配到你手里的事儿做出来的能力——活干出来了，这是个结果。“扛事儿”则指的是你干活的过程里，表现出的态度和品质，这其中最重要的，就是 **Owner** 意识。

举个反例，是一个之前试用期就翻车的同事。这位前同事入职初期接了一个系统，注意力从头到尾都 **focus** 在“这个功能怎么做”、“那个脚本怎么写”上，完全没有 **care** 这个系统开发过程中潜在的风险，也没有感知某一个功能点背后不合理的业务逻辑，和产品经理、运营等多方的沟通也存在不同程度的滞后，最后团队其他人不得不跑过来给他擦屁股，进而影响了整个团队的人力调度和资源分配。在这个过程中，虽然前同事也没少干活，但也没少给别人添麻烦。这种情况，在大厂严格的流水线作业分工模式下，是非常忌讳的。所以说，**Owner** 意识非常非常重要。

Owner 意识，如何在面试时体现？一样，看项目。在描述自己做过的项目时，要有头有尾地说清楚：为什么要做这个东西（背景、对业务的理解）？做的时候怎么想的（方法和选型）？做的过程中遇到了哪些问题，是怎样解决的（独立思考能力）？做完之后有哪些反思（复盘能力）？站在一个完整的研发流程的视角去讲你的故事，而不是单纯站在技术工人的角度去片面地陈述技术细节。简言之，要有**全局观**。

}