

50 功夫在平时：写给长线作战的同学们

更新时间：2020-06-23 15:38:50



“ 一个不注意小事情的人，永远不会成功大事业。——戴尔·卡耐基 ”

对于前端面试，笔者个人推崇的策略是“长线作战”。短期突击除了难度大以外，还会带来“面完就忘”的副作用，不利于工程师的可持续发展。基于此，我在这里给长线作战的同学一些建议。

面向出身普通的校招同学：做一些能够“留下痕迹”的事情

如果你是非名校/非科班的在校生，学历对你的钳制相对会多一些，因此必须要做一些能留下痕迹的事情来证明自己有着足以覆盖学历劣势的学习能力和实践经历，具体的做法有：

- 在校期间抓住一切可以实习的机会，寒假暑假、甚至日常实习。如果没有公司接收，那就去社团、去技术 club，总之要将工作经验的进度条提前，积累生产级别的项目经历。校招时，别人是学生弟，你已经算半个经验工程师了。
- 建立自己的博客和github，持续输出对技术的思考、主动寻找建立/参加有价值的开源项目的机会。这些东西不仅仅帮你构建自己的学习历史，它们被 po 上简历后的杀伤力是你所想不到的。如果你对博客搭建毫无经验，那么推荐你从 [Hexo](#) 开始。
- 在有分量的大赛中拿奖——这里的“有分量”是指国家级及以上的计算机类大赛。省级及以下的比赛，如果具备较强的技术性，并且能够确实产出你的技术作品，那么也可以一搏。总之，名次和作品，至少要有一个拿到结果。
- 对不具备学历优势的校招同学来说，上述三种途径是效果显著的积累大厂敲门砖的办法。笔者观察了三年校

招，双非候选人拿到 offer 的，基本都做到了上述三点中的两点，无一例外。

面向在职工程师：每做一件事，都要思考这件事和简历之间的关联

拿到一份社招简历，第一眼，会看他之前是否出身大厂。如果是，那么你通过简历评估、拿到一面的机会就稳了九成。

如果没有大厂背书，面试官在评估时就会详细地过一遍你具体做过哪些事情。这种情况下，你需要靠谱的业务经验&研发专项作支撑，基于此，各位当下可以采取的行动有：

以业务为突破点：

就从眼前的团队、眼前的业务下手——先捋清楚一个问题：在手头这个项目/业务里，我承担了什么样的角色？是不足轻重，还是技术中枢？如果眼下只有拧螺丝钉的余地，那么能否争取独自 owner 一条业务线/一个项目的机会？

这样的机会非常重要！owner 完整业务的过程，其实是在间接地刻意练习自己对业务的理解能力、对技术选型的思考以及把控整个项目上下游风险的能力。有的同学可能会对此不屑，认为前端只需要关心前端实现，不需要操太多“闲心”——这也是我想要在此矫正大家的一个比较重要的普遍的认知误区。我的建议是，不要把自己定位成施工队队员，把自己定位成项目的 PM，主动去操一份“闲心”，把业务的整条链路——从算法召回数据，到服务端分发数据，再到前端/客户端渲染这个过程捋清楚。不必强求细节（很多跨工种的细节本身理解成本也够高了），但要搞清楚这个流程里都有哪些动作以及这些动作和自己所在这个环节的关联。操这份“闲心”，带来的一个最直观的首先是你不容易背锅了。

举个例子：我们来看一个性能优化的场景。现在你负责了一个页面，产品说它启动太慢了，需要查下原因。一查之下，你发现它首屏要加载 3 个接口的数据，你的第一反应是什么？（说“与我无关”的同学，我要打人了。。。：））。一个比较本能的反应应该是，“3 次请求拖了我页面的速度，服务端为什么不能合并到一个接口里？”。这时如果你对整个链路有所了解，你知道服务端的系统哪些是互通的、哪些是隔离的，是不是就可以理所当然地要求他们合并接口了？

退一步想，即便服务端没法改，请求过程中的性能确实要做些让步，那么如果我们对客户端处理前端资源的方式比较清楚，是不是也比较容易发现可操作的点？（比如客户端容器可以提前帮我们吧前端资源缓存到本地，减少加载过程耗时）。如果你对整个链路的上下游是清楚的，你就可以很自然地要求两端的同学做出一些有利于页面性能的改变（反之如果你是不清楚的，就只能接受页面频繁白屏和卡顿的结局）。经验告诉我们，这些看似“与前端无关”的动作带来的改变，往往是前端页面性能质的飞跃——比我们自己埋头不停地优化 DOM 效果显著多了~

当然啦，“不接锅”只是短期利好。长期来看，我们的终极目标是把这样的一个独当一面、协调各方的经历写进简历，并且在面试环节一板一眼地说给面试官听——以社招的身份进入大厂，能够独当一面，是业务能力的底线。

以技术专项为突破点：

没有机会/能力做业务 owner，怎么办？如果是技术不牢靠的问题，可以先跳至“硬实力”板块做技能树对齐。此外，学会变通，学会利用眼前你所能触及到的资源。举个例子：我目前只能负责一小块业务，甚至只是一个页面，那么这个页面就是我的机会。代码如何重构，才能具备更强的健壮性？应用哪些手段，才能让页面的性能达到“秒开”，让用户体验如丝般顺滑？——以这些具体的问题为切入点，利用这一个小小的页面去拓展自己的技术攻坚能力，发展自己的技术专项——哪怕，只是为了把它写到简历上。

创造从 0 到 1 机会：

做一些能够影响整个团队的事情，会为你的过往履历写入浓墨重彩的一笔。做到这一点，未必需要你特别强，但一定需要你特别留心。具体来说，尝试去观察现有团队存在的问题：比如，有哪些工作，是正在被大量重复的？一类组件被反复写了 N 多次，那么我是不是可以推动团队去建立自己的组件库？前端代码的构建发布流程非常繁琐，那么是不是可以尝试推动团队的工程化？提出自己的想法，把它落地成具体的工具/库，是 owner 能力和技术攻关实力的双重体现。

补齐技术短板：

关注自己的弱点。大部分前端工程师是野路子，不具备扎实的计算机科学基础、抗拒理论学习、高度依赖实际经验。有人会以此为荣，但希望你不要这样做——保持清醒，在技术领域，一样知识会总比不会强，更何况是作为学科基石存在的这些理论知识呢？编译原理之类的若不必深究，那么软件工程、设计模式、计算机网络和算法这些工程领域的理论基础也应该引起你的重视。

随着前端应用的复杂化、前端工程师 Job Model 与软件工程师 Job Model 渐渐趋同，这些知识渐渐从面试爱问不问变成了面试必问，已经成为前端工程师硬实力的重要组成部分。

写在最后

50节，10w+字，这是我写过的最长一个专栏——非常感谢能够读到最后一节的同学们！

经过长达三个月的学习，相信各位已经逐渐对前端面试所涉及的知识体系及世界观都形成了自己的理解。在结束了以面试为导向的学习后，笔者希望大家能够思考这样两个问题：学过的这些知识，我应该怎样处理？明白了前端面试的评价逻辑后，我在平日的工作中应该朝着怎样的方向努力？

对于一个前端工程师来说，面试成功、加入一个更好的团队，是变强路上的重要环节，但绝不是他奋斗生涯的终点。相反，当你站在这个平台上后，才真正是棋逢对手、需要拿出看家本领去搏一搏的时候。因此，保持对技术的热情和严谨，应该是一个一以贯之的过程；保持对学习的热忱和耐心，也应该是一个一以贯之的过程。

最后，希望各位都能有美好未来。如果这份专栏能在这个过程中帮到各位，也是我的幸事。

大家加油！

}



49 临门一脚：前端面试的本质是什么？