

开篇词 | 为何掌握了技术API，依然在项目中处处掣肘？

2022-11-20 杨文坚 来自北京



《Vue 3 企业级项目实战课》

[课程介绍 >](#)



讲述：杨文坚

时长 13:03 大小 11.92M



你好，我是杨文坚，欢迎跟我一起学习 Vue 3 的企业级项目开发。

从中山大学毕业，我一直在从事前端开发的相关工作，期间就职过腾讯和阿里巴巴等互联网头部企业。在过去十年的职业生涯中，我担任过一线前端开发、项目经理和技术主管，带领过规模约二十人的前端技术团队，涉猎了多个技术方向，包括自研前端模板引擎（MVVM 框架）、中后台一体化（微前端相关）、多端框架融合（React.js 语法转换）、前端页面定制化搭建（低代码）、前端错误监控和前端基建，等等。我也曾受邀成为 D2 技术论坛的技术分享嘉宾，担任过 StuQ 和极客时间的技术培训讲师。

在 StuQ 和极客时间做技术培训讲师期间，我发现有这么一个现象。当出现一个新的技术框架，或者某个技术框架升级后，大厂里的同事往往能很快上手，并且将它运用到实际项目中。同一时间里，我所带的技术培训班级的学生虽然也能基于框架的官方文档，快速上手新技术，但就是不知道如何将它运用到实际项目中，或者不清楚还有哪些用途。

我就非常奇怪：同样是能快速上手新技术，但是两者在具体项目中运用新技术就天差地别，难道真的是能力上的差距？



在很长一段时间里，我不断跟大厂同事探讨这个现象，同时也会跟班里的学生沟通技术运用上的“阻力”。后来，一些同事的话让我恍然大悟，大概是说，“这无关技术框架的新旧，其实就是按照以往经验来用新技术”，还有人说，“在当时这个项目背景下，只有选择这个新技术才能解决”。

我把这些话总结成了两个要点，一个是“术”和“道”的差异，另一个是“环境”的差异。

“术”与“道”的结合

什么是“术”和“道”的差异呢？这里说的“术”是指具体某种技术框架的使用，例如 Vue.js、React.js 这类具体技术框架的 API 使用。“道”是指针对某个具体问题，在技术视角的解决思路是什么，例如要完成一个页面搭建平台项目，就要解构这个平台的整体内容，要先分析平台的逻辑链路、设计技术分层、设计技术架构等。

而所谓“环境”的差异，指的就是大厂里的项目机会。大厂里经常会有创新性的项目需求，而且绝大多数项目都是必须实现的，甚至项目的落地与否跟年终绩效和末位淘汰挂钩。在这个“环境”和压力的影响下，大厂的程序员不得不想方设法，掏空自己的技术知识储备来实现项目的需求。如果知识储备不够，他们就得八仙过海、各显神通地往新技术上寻找解决办法。

反过来这个“环境”也提供了一个实际的舞台，让大厂程序员能充分利用实际项目验证这个新技术是否可靠。这时候，大厂程序员能从实际项目中得到新技术的运用经验，从新技术框架的 API 使用中（“术”的运用）理解这个新技术在项目中的哪个逻辑、哪个环节发挥了什么作用（“道”的沉淀）。

通过这些分析，我发现两者新技术运用差异的本质并不是个人技术能力上的差异，而是非大厂程序员**没有“大厂级的项目环境”来沉淀“大厂级的项目经验”**，最后无法达到“术”和“道”的相辅相成的螺旋式成长。

正是因为有这样大厂的工作经验和技术培训讲师的教学感悟，我萌生了要把大厂里做项目的“道”融入专栏的“术”里的想法，让非大厂的前端开发者也能感受大厂的开发“环境”，从而不断提升自己对开发的“道”的理解。

那么，用什么“术”作为课程载体来教学“道”呢？

我认为，“术”的选择必须是主流的技术框架，并且是能满足前端就业市场需求和个人能力提升需要的技术。在前端就业市场里，很多企业招聘要求都是“熟悉 **Vue.js** 和 **React.js**”，但实际上，**Vue.js** 的招聘岗位比 **React.js** 要多一些。而且在技术深度上，两者发展日益趋同，**Vue.js** 最新版本 **3.x** 就融入了很多 **React.js** 的技术概念，例如 **Vue.js** 的 **Composition API** 是借鉴了 **React.js** 的 **Hooks API** 的理念。

所以，最终我结合就业市场和技术深度两个核心因素，选择了 **Vue.js 3.x** 这个版本的技术来作为课程内容的“载体”。我将借助这个“术”，给你分析和讲解学习和掌握企业级项目的技术之“道”。由此也就引申出了非大厂程序员经常会问的一个问题：为什么熟悉 **Vue.js** 官方 **API** 文档，还是用不好 **Vue.js**？

为什么熟悉官方文档，还是做不好项目？

我知道很多人都是带着问题来学习这门课程的，你可能想问，**Vue.js** 如何打造自己的组件库、**Vue.js** 如何做搭建功能、**Vue.js** 如何结合 **Node.js** 做全栈项目，等等。

这些问题固然重要，但如果你一直停留在基于 **Vue.js** 解决琐碎的功能问题这种所谓“术”的阶段，只会关注一些 **API** 的使用，就无法形成一个属于自己的技术知识体系和解决问题的方法论，无法踏上我们前面一直在说的“道”的阶段。

我来举一个实际的例子，假设你接到了项目任务，需要你负责开发一个问卷调查平台，并且整个项目只有你这一个程序员负责，你会怎么开展研发工作呢？单纯查看 **Vue.js** 官方 **API** 文档是找不到解决方案的，我给你分析一下如果实现这个“调查问卷平台”项目，需要有哪些执行步骤。

第一步，需要梳理功能逻辑。我们要确认这个调查问卷平台需要哪些功能支持，这些功能的流程操作顺序是如何布置，然后用文字或流程图给整理出来。

第二步，设计技术方案。梳理完功能逻辑后，可以根据功能逻辑的需要来选择和定制技术方案。例如问卷调查表单的实现，比较理想的技术方式是“动态表单”。但只有一个前端开发如何实现后台功能呢？最快速的方法是利用前端熟悉的 **JavaScript** 语言，基于 **Node.js** 来开发这个平台的后端服务。把这些功能逻辑逐一分成一个个技术点，再找对应技术点的具体实现方式，整合出一套技术方案。

第三步，开发落地功能。根据上一步里的技术方案设计，将其中的一个个技术点逐一开发写代码实现功能，最后直至所有功能全部开发完毕。



第四步，规划后续平台维护。功能代码开发完了不是一个项目的终点，项目的终点是结束历史使命下线的时候。在项目开发完成到项目下线的中间过程，我们还需要制定整体的代码维护规划，例如出现问题要如何迭代，如何增强代码的日志监控来提升排查问题效率等等。

通过这个例子，相信你已经了解到，在实际的企业级项目中，单纯依赖 Vue.js 的 API 技术知识是远远不够的，你需要的是一整套围绕 Vue.js 建设项目的技术知识储备体系。

而大部分程序员学完 Vue.js 的官方 API 文档后，很难进一步做好项目，核心的障碍就是缺少一个属于自己的技术知识储备体系，以及对这个知识体系的利用技巧。“技术知识储备体系”和“利用技术知识”就是上述所谓“道”的具体实际内容，这也是我希望能够通过这门课程带给你的。

所以，在这门课中，我会围绕着实用主义来辅助你沉淀“技术储备”，再以一套有持续性的项目实战来辅助你学会“利用技术储备”。那么，我是如何设计这个课程呢？

我是如何设计这个课程的？

总的来说，我们这门课会以“大厂企业级项目”为大背景，主要围绕“前端低代码”和“页面搭建”这两个当下大厂热门的技术项目方向，最终目的是手把手教你从零实现一个“运营搭建平台”项目。

在这个项目的准备和实现过程中，你会学到 Vue 3 自研组件库开发、Node.js 服务端开发、Vue 3 结合 Node.js 的全栈项目开发以及全栈项目的单元测试等大厂常见的项目技术场景。在此期间，你可以沉浸式地体验到大厂开发项目的全过程，从中积累属于自己的技术知识储备。

具体来说，分成以下基础篇、进阶篇、实战篇和增强篇这四个板块，下面我们简单介绍下。

- **基础篇：换个视角，带你从“实用主义”来重新入门 Vue 3**

无论你是否学过 Vue 3，我都建议你跟着我通过实用主义的视角重新认识一次 Vue 3。在这个过程中，我会列举一些常见的项目需求，分析 Vue 3 有哪些对应方式可以作为备选方案，再因地制宜地进行选择。

例如，我们如何解决浏览器里动态组装 Vue 3 组件的问题呢？我会告诉你 Vue 3 组件是如何以非编译的形式在浏览器里运行的，以及多种非编译方式的优劣特点，最后再扩展问题面发散思考，分析这类技术方式以后还能应用到什么场景之中。



如果你之前学过 Vue 3，你可以通过这部分内容的学习，对自己 Vue 3 的基础重新回炉淬火。在这时候，你会摆脱“只会官方 API 文档，不知如何实际运用”的窘境，让快被“尘封”的知识点重现在聚光灯下，让自己快被“尘封”的技术武器重新“淬火”，变得更加坚硬和坚韧。

如果你之前没学过 Vue 3，也不用担心，你可以在这部分直接学到如何针对具体问题“对症下药”，选择合适的 API 来解决问题。

• 进阶篇：沉淀方案，循序渐进地打造自己的技术知识储备

经过对 Vue.js 的基础知识的“加固”后，这一阶段，我们将来学习大厂里企业级项目会用到的技术知识点。这里讲的知识点，不仅包括了 Vue.js 的生态知识，更多是前端领域的一些技术概念。同时，我们也会来学习如何来理解这些概念，如何用 Vue.js 的能力实现这些技术概念。例如动态表单、单元测试等前端技术概念如何在 Vue.js 技术中得到实现，等等。

在这期间，我们会将常见的大厂项目里的技术概念转换成具体场景问题来实现，让你可以逐渐沉淀属于自己的技术知识储备，编织属于自己的技术知识脉络体系。

• 实战篇：厚积薄发，因地制宜设计技术方案和实现技术功能

经过上个阶段技术知识的积累，我们就相当于厨师有了菜谱和食材，接下来要做的就是根据顾客的需要来打造一顿技术盛宴了。

在实战篇阶段，我将带你，将低代码搭建页面的技术概念融入到课程中。我们会化整为零，将一个企业级项目分解成一个个简单易懂的技术知识点，也会因地制宜地用合适的实现方式解决一个个技术问题点，从而让你知道如何优雅地利用自己的技术知识储备解决实际的项目问题。

• 增强篇：打破边界，技多不压身，解锁前端以外的技能

在最后的增强篇，我将带你深入 Node.js 服务端领域，了解如何打造一个稳定的 Node.js 服务生态。例如，如何给 Node.js 做多进程和进程守护来提升服务性能和稳定性，如何快速定位

Node.js 服务问题原因和解决问题，等等。

通过这一部分的学习，你可以贯穿理解前端开发、服务端开发和技术运维的内容，理解什么是“开发即运维”，逐步掌握这一大厂程序员的技术“硬实力”和应对突发问题的“软实力”。

完课之后，希望你在面对大厂级别的项目的时候，能够“独当一面”地去披荆斩棘解决问题，乘风破浪取得成果。

具体内容，你可以看下面这张目录图：

《Vue3 企业级项目实战课》课程目录

■ 开篇词 | 为何掌握了技术 API，依然在项目中处处掣肘？

基础篇

01 编译和非编译模式：离开 Vue 工具，你还知道怎么用 Vue 3 吗？

02 Webpack 编译搭建：如何用 Webpack 初构建你的 Vue 3 项目？

03 从 Rollup 到 Vite：如何用 Rollup 构建你的 Vue 3 项目？

04 模板语法和 JSX 语法：你知道 Vue 可以用 JSX 写吗？

05 响应式开发操作：如何理解和使用 Vue 3 的响应式数据？

06 跨组件数据通信：常见的组件间数据通信方式有哪些？

07 项目代码规范：如何成为一名合格的团队协作工程师？

进阶篇



天下无鱼

<https://shikey.com/>

08 如何从零搭建自研的 Vue 组件库？

09 主题方案和基础组件：如何设计组件库的主题方案？

10 动态渲染组件：如何实现 Vue 的动态渲染组件？

11 布局组件：如何实现自研组件库的布局方案？

12 受控表单组件：如何开发受控的表单组件？

13 动态表单组件：怎么优雅地动态渲染表单？

14 代码单元测试：如何轻松地保证自己的代码质量？

实战篇

15 定制运营拖拽组件：如何实现运营搭建页面的拖拽功能？

16 单页面应用：如何理解和实现单页面应用开发？

17 Koa.js：如何结合 Koa.js 开发 Node.js Web 服务？

18 Node.js 服务端渲染页面：客户端渲染和服务端渲染有什么区别？

19 全栈项目搭建：如何搭建 Vue.js 的前后台全栈项目？

20 数据库方案设计：如何设计运营搭建平台的数据库？

21 用户注册和登录：如何结合 Vue 3 和 Koa.js 实现注册登录？

22 物料组件的编译和管理：如何处理组件的多种模块格式？



天下无鱼

<https://shikey.com/>

23 运营物料的后台管理：如何全栈化实现列表分页的功能？

24 后台搭建数据源：如何设计运营搭建页面的数据结构？

25 后台搭建功能：如何设计和实现 Vue.js 运营后台的搭建功能？

26 前台页面渲染：如何设计 Vue.js 搭建页面的渲染策略？

27 后台发布流程：如何实现 Vue.js 搭建页面的发布流程？

28 前台页面版本化管理：如何实现搭建页面的迭代策略？

29 前台页面的渲染方式：如何设计前台页面的渲染策略？

30 前台页面的性能优化：如何实现前台页面的性能优化？

31 前台页面的日志监控：如何进行页面实时监控与问题定位？

增强篇

32 如何打造 Vue.js 和 Node.js 全栈项目的单元测试？

33 页面功能扩展：如何对 Vue.js 全栈项目做优雅的扩展？

34 服务端功能扩展：如何对 Vue.js 全栈项目做服务端功能扩展？

35 多进程部署：如何最大限度利用服务器资源运行 Node.js

服务？

36 日志收集与问题排错：如何守护好 Vue.js 和 Node.js 的全栈项目？

<https://shikey.com/>


■ 结束语 | 时刻做好“军备竞赛”，机会只留给有准备的人

最后，我想跟你说，Vue 3 是个非常优秀的技术框架。但是光学会 Vue 3 是远远不够的，最终我们需要以 Vue.js 的使用作为学习线索和项目方案路径，打造一个属于你自己的技术知识储备体系。并且，通过这些新技能的学习，打破前端的边界，往全栈开发领域进击。

另外，在这里你也不需要担心自己基础不好跟不上，我会把课程里所有的代码都放在公开的 [GitHub 仓库](#) 中，你可以自己慢慢琢磨，一点一点尝试。有任何问题，欢迎你在评论区多多留言，我们一起解决。

那么，我们课程第一讲见！

分享给需要的人，Ta 购买本课程，你将得 18 元

 生成海报并分享

 赞 3  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。

下一篇 01 | 编译和非编译模式：离开Vue工具，你还知道怎么用 Vue 3 吗？



Vue3 企业级项目实战课

进阶高手的 Vue3+Node.js 全栈开发训练

杨文坚

前阿里前端 leader

前腾讯 IMWeb 团队高级前端工程师



新版升级：点击「 请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。

精选留言 (4)

写留言



WGH

2022-11-21 来自陕西

十几天前就在11月上新课表中关注啦，今天终于更新啦。

开篇词中提到的围绕Vue打造自己的技术知识体系，单元测试，特别是大厂沉淀下来的场景经验，都是我欠缺的。非常期待这次学习。

为极客时间打call，为作者打call。



2



十九

2022-11-22 来自北京

来了来了



徐曙辉

2022-11-21 来自湖南

对于服务端开发者能学到哪些知识点？



风太太太太

2022-11-21 来自湖北



天下无鱼

<https://shikey.com/>

前半部分讲的很好，支持老师一波，立个flag，坚持周一 周三 周五上班时间摸鱼，关注订阅老师的课程。

