

结束语 | 时刻做好“军备竞赛”，机会只留给有准备的人

2023-03-10 杨文坚 来自北京

《Vue 3 企业级项目实战课》

[课程介绍 >](#)

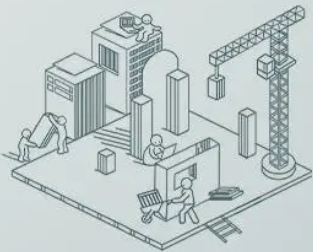


杨文坚

前阿里前端 Leader, 前腾讯 IMWeb 团队高级前端工程师

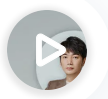
你好, 我是杨文坚。

作为程序员, 我们的工作主要是用技术方式解决实际问题, 不要把自己局限在前端或者服务端等单一技术领域, 跳出现有领域“山重水复”的阻隔, 你才能发现“柳暗花明”的新世界。



讲述: 杨文坚

时长 09:06 大小 8.32M



你好, 我是杨文坚。

首先恭喜你, 因为你的坚持和努力, 你已经完成了整个课程的学习。

我们用三十六节课, 围绕着“运营搭建平台”这一项目命题, 构建了一个企业级项目的方案体系, 在学习前端、后端和系统运维的技术知识点之余, 也掌握了实现企业级项目的设计思路以及不同开发问题的应对套路。

根据课程的整体脉络, 我们先梳理一下要构建企业级项目需要掌握的知识体系。

梳理课程的整体脉络

对于整个课程来说, Vue.js 3.x 只是一个“引子”。大思路是, 我们首先讨论如何用它来解决实际的需求问题, 然后根据实际项目场景, 由浅入深把项目分解成独立的需求问题, 对问题点逐

一设计方案，最后实施落地。

在“基础篇”中，我们先回炉学习了 **Vue.js 3.x**。但不同于枯燥的框架 API 解读，我们以工作中常见开发的场景为例，按需分析每个技术该怎样使用，比如 **Vue.js 3.x** 的常规使用方式、编译方案选择和项目开发规范管理。

核心理念是**把握有限的时间，快速入门或者重学技术的实际用途**。

然后我们进入“进阶篇”，从企业常规的前端组件需求出发，从零开始搭建 **Vue.js** 组件库，并对组件类型进行归纳处理。这个部分，我们进一步扩展需求场景的范围，把典型需求场景归纳成具体技术问题，再从 **Vue.js 3.x** 的技术特性里，整合出解决方案，封装成组件，解决问题。

核心理念是**为了解决问题而学技术，而不是为了学技术而学技术**，避免陷入“已熟悉框架 API 文档，还是做不好项目”的困境。

在“实战篇”中，我们基于 **Vue.js** 和 **Node.js** 构建了一个全栈项目，并且把技术知识点贯穿了前端、服务端、数据库等领域。核心需要掌握的有两点：**如何构建全栈项目的技术体系、大型项目的功能分析思路**。

相信你从搭建平台的“功能分点拆解”，以及“功能点实现”的方案套路中，能构建出属于自己的项目方案套路。

最后是“增强篇”，我们不能纯粹地学技术或者方案，而是要跳出“功能实现”的刻板技术思维，提升到项目“功能维护迭代”的项目管理思维。核心要明白，**企业项目不能停留在“做到功能”的程度，而是要时刻保持“做好功能”的态度**。

其实，我们反复提到了“解决问题”和“学习技术”这两个概念，你可能会疑问，如果说“不能单纯为了学技术而学技术”，但是想“解决问题”又要依赖“学习技术”时储备的知识，那应该先“解决问题”呢？还是应该先“学习技术”呢？

先解决问题？还是先学习技术？

其实，这就跟“先有鸡还是先有蛋”类似，没有绝对的答案，也没有绝对的先后顺序。因为“解决问题”和“学习技术”是相辅相成、螺旋上升的过程。

根据之前的工作经历，我的看法是：“**在解决问题中学习技术，在学习技术中解决问题**”。

“在解决问题中学习技术”，是因为我们工作中遇到的问题，不可能每次都有现成的技术方案，也不可能自己都知道怎样解决。遇到问题，在时间的压力下，我们大多数情况都是查找技术现学现用，通过学习和实践来找到解决方案，这个过程也是在学习技术。

“在学习技术中解决问题”，是因为我们学习技术不能“教条主义”地学，而是要设想问题场景或者实用场景。比如某天 Vue.js 出了新版本 4.0，我们不能为了追潮流，学习 4.0 的新技术，而是要理解 4.0 解决了什么问题、能带来什么实际生产作用、整合出什么新技术方案、能带来什么收益。

如果你在学习技术的时候，能和问题场景关联起来，学习和使用会更加深刻，反之不仅会比较吃力，学后也很容易遗忘，这也再次回答了开篇词中很多人的疑问——“[🔗为什么熟悉 Vue.js 官方 API 文档，还是用不好 Vue.js 做项目？](#)”。

总的来说，两者是相辅相成的，不区分孰先孰后，核心理念就是从“**实用主义**”出发，发挥技术的“**实际作用**”解决问题。

不过，无论是“解决问题”还是“学习技术”，我们最终都要沉淀出属于自己的“技术知识储备”。

构建属于自己的技术知识储备

“技术知识储备”这个词有点太理论了，我们换个说法，构建自己的“技术知识储备”，说白了就是**持续积累解决问题的经验，形成自己在开发工作中解决问题的惯用思路**。

那么如何沉淀解决问题的经验，我总结了三点。

- 归纳技术特点
- 分析功能场景
- 整合方案设计

第一点，归纳技术特点，就是指梳理纯粹技术点。**根据每项技术的特点，归纳出有什么实际作用**。

相信你在工作中或多或少专门看过某项技术的使用文档，比如课程搭建平台里用到 JSON Schema 技术，就是纯规范的概念，作用是“用 JSON 来描述 JSON 格式”，我们就可以把它归纳成“用于描述 JSON 数据格式的技术”。

第二点，分析功能场景，基于自己学习过的技术内容，**根据当前的功能场景，先分析功能类型，再根据功能类型的需要，选择自己熟悉的技术点来解决。**

比如课程搭建平台的物料配置功能，因为不同物料组件的属性格式都不一样，配置功能需要动态渲染不同组件的表单。这时候，功能类型，我们可以划分成“动态数据的动态表单渲染”，而“动态数据”需要用描述的数据格式，“动态表单”也需要描述数据格式，我们就可以用已学的“JSON Schema”这项技术点来实现。

具体就是在 Vue.js 组件的 TypeScript 类型声明文件中，把组件属性类型转成 JSON Schema，基于 JSON Schema 生成动态表单的字段配置，渲染出动态表单。

所以，对于“动态数据的动态表单渲染”的功能类型，我们一共要用到三个技术点：JSON Schema 规范、TypeScript 类型转 JSON Schema 技术、动态表单渲染技术。以后遇到类似的功能点需求，我们就知道如何选择技术点来实现了。

第三点，整合方案设计，**面对一个完整的项目需求，先拆解成一个个独立的功能点，然后基于自己所积累的功能场景，整合成一套完整的项目技术方案。**

如果所有技术点和功能场景，你都能游刃有余地分析出来，就说明你发挥了自己技术知识储备的作用，这种沉淀了丰富经验后解决问题的“惯用思路”，也就是我们说的解决问题的“套路”。

“套路”积累得越多，我们遇到项目问题或者技术问题，才能条件反射地想到解决方案，之后应对企业内部晋升答辩，或者是跳槽技术面试，都能游刃有余地应对晋升评委或面试官的各种挑战。

不过，有了一定程度的技术知识储备，是不是意味着我们能应对所有工作场景呢？

如何度过个人发展的瓶颈期

我个人认为答案是否定的。毕竟，技术知识储备只是解决技术问题或需求问题，实际情况下，我们程序员的职业生涯除了技术问题，还有很多非技术的问题，比如项目管理、合作沟通、职

场晋升、就业跳槽等等。

有了技术知识储备，相当于是军人在战场中有了“战术储备”，解决的是“战术问题”。个人瓶颈期是一个“战略问题”，涉及大环境、就业发展趋势等等，这些不是我们单凭个人努力就能左右的，但是，**这些非技术问题，才是我们个人发展的瓶颈期问题。**

这时候，我们能做的是尽量把“战术储备”转化成“战略储备”，应对“战略问题”，来度过瓶颈期。

在现实工作中，当我们的个人技术能力能覆盖大部分项目问题的时候，具体的瓶颈期问题就是环境问题，比如互联网就业环境遇到寒冬、企业发展不利一直晋升不了等等。这也是我职业生涯遇到过的。


我自己的做法是把技术知识储备转化成行业发展储备，拓宽自己的职业发展方向可能性。不一定要死守在互联网行业或者江河日下的企业里，我们还可以用互联网技术思维，去解决其他行业的问题，在其他行业里找到就业或者个人发展的新方向。

当你经历了这些过程，也做好了充分准备，大部分问题都将不是问题，剩下的，就是等待机会的到来。

最后在专栏要结束的今天，我也非常希望能在留言区见到你的身影，听听你学习这个专栏的感受和收获，也期待你提出学习课程的反馈与建议（戳这里填写 [📄 问卷](#)）。

祝愿你在以后的工作学习中，顺利跳出“瓶颈期”的“山重水复”，逢山开路，遇水搭桥，直达新世界的“柳暗花明”。

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 1  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。

精选留言 (1)

写留言



Akili

2023-03-16 来自云南

谢谢老师

