

前端工程化

> 波比

内容：

- 前端工程化概述
- 某厂工程化案例介绍
- 手写迷你打包工具
- 提问环节（不限于本节课内容）

前端工程化概述

从学科角度，最接近的概念就是「软件工程」

从文字角度来看，分为两部分，「前端」、「工程化」，分别指的是「大前端方向」「系统级的体系建设」

从流程上看，就是「编码」「测试」「构建」「发布」「运行」「维护」的一整套体系

...

> 很难去给定一个清晰的描述，多是根据经验总结

>

> 我所理解的前端工程化就是结合公司现有环境（基建、人员、产品形态等等），提供开发全生命周期的一整套开箱即用的解决方案体系

我们从流程的角度去细化，会发现「前端工程化」是一个非常庞大且复杂的体系，就算看起来我们罗列出如下这么多内容，依旧不能代表前端工程化的所有内容

「编码」

就是开发，这个阶段往往面临的是选择题

假如你接手一个新的项目，你会考虑到的东西有哪些？

- 技术选型，用 React 还是 Vue，用 Koa 还是 Nestjs？
- 开发模式选择，前后端完全分离？同构直出？
- 组件库选择，Antd？
- Mock 方案，PWA mock？Node Server Mock？MockJS？
- 全家桶，路由？状态管理？调试工具？
- 脚手架，手撸？CRA？VueCLI？
- 目录划分
- 领域模型设计
- 分支管理方案
- 微前端
- ...

往往在你编写第一行业务逻辑或者页面代码之前，就得经过这么一系列的思考，这个时候有同学会问，直接 Umi 一把梭哈不就行了？

> 那么这个 Umi 这样的集成解决方案，不就正是一种前端工程化嘛

「测试」

很多外企喜欢 TDD，毫无疑问，规范、严格的 TDD 模式下能产出非常鲁棒的代码

测试也是工程化中重要的一环，PS：很多公司目前都不太能做到开发过程中有完整的单测覆盖，主要还是成本问题，依靠的还是提测之后 QA 环节的质量把控

所以测试，我们可以分为两部分，第一部分是开发中的，由 FE 同学自行实现的「性能测试/优化」「代码规范」「单元测试」「E2E」「覆盖率」等测试体系，这些在现在的前端生态都有足够的开源工具帮助实现

第二部分就是到 QA 环节提供的测试能力，「手动测试」「冒烟」「AB」「自动化测试」等等，这里也有可以通过技术手段进行自动化，工程化实现的点

「构建」

发布其实是有两个动作，一个是「构建」，提到这个词，大家脑中一定闪现了如下词汇「webpack」「rollup」「vite」「配置工程师」...

现在我们最常使用的应该是 webpack，想想它直接或者间接给我们带来了哪些能力？

- 依赖打包（模块化）
- 文件压缩
- code splitting
- cache
- 增量更新
- 热更新
- CSS 预处理或者后处理
- Build bundle or chunk
- 资源定位
- ...

有很多同学会认为 webpack 就是工程化了，其实是不对的，webpack 只是在系统化、工程化的解决诸如上述问题的时候总结出的一套解决方案

「部署」

部署简单说就是把代码放到 CDN、机器这些地方去，「运行」，「维护」其实也是一起的，所以我们一起说了，很多前端同学在这一 part 往往不够重视，我们先梳理一下这部分涉及的内容有哪些？

- 持续集成/持续交付（CI/CD）
- Nginx
- Docker、K8S
- Shell
- 灰度
- HTTP（S）
- 网关
- HUB
- 监控
- Jenkins
- ...

这些东西其实就是所谓的基建了，很多年头比较久远的互联网公司都有对应的配套基建，提供了一些可视化、命令行工具等开箱即用的解决工具，比如百度的 ICode

现在很多同学喜欢研究一些”符合时代“的方向，比如「性能优化」「跨端」「low/no code」等，「前端工程化」领域并不受重视，搞一个脚手架，集成了所有你能查到的 webpack 最佳实践，并不是「前端工程化」了，但是这是一个很好的开始，如果顺着这条路发展下去，在这个领域成为专家，也是叩开众多大厂大门的好途径

「前端工程化」对开发来说，可能就是懒，希望一系列工作可以集成，哪怕是黑盒，对公司来

说，则是降本提效的重要手段，总之，老板喜欢，个人有成长，这个领域的竞争也小，是个蓝海市场（手动狗头）

> 搞工程化，不是为了提升技术而搞，最根本目的是服务于业务，不能促进业务发展得更好，啥技术都是扯淡 -- chengxi li

实践路线：

成本最小的就是基于现有的社区中优秀的集成解决方案进行改造，比如 UMI，因为 UMI 本身设计有插件系统方便拓展，按照我们上边总结的（结合实际增删内容）有耐心的改造它，最后就会形成一套工程化的解决方案，可以在你司推广大家使用，再持续迭代优化

- Fork 一份 umi 源码
- 结合实际更换内容，比如组件库改成你司自己的，路由库换成自己的等等
- 结合实际增加部署 CDN，部署到机器，Nginx 转发，甚至可视化部署等能力
- 接入监控，SEO 优化等等 UMI 没有的能力

某厂工程化案例介绍

我们以从开发到部署全流程一个 Node 服务为 Case 介绍一下某厂的工程化包含哪些内容，让大家有一个直接的感知认识。