前端工程化实践

课程大纲

⚙️ 基础篇

- Git 版本控制
- Npm 包管理器
- 脚手架与 CLI
- 模块化
- 组件化
- 开发规范
- Rollup 基础
- Babel 语法编译器
- ESLint
- Prettier
- Polyfill 垫片与浏览...
- Jest 单元测试
- 调试工具
- Vite 基础
- Webpack 基础
- 代码覆盖率
- 持续集成 CI/CD

🔎 原理篇

- mini-webpack
- mini-vite
- mini-rollup

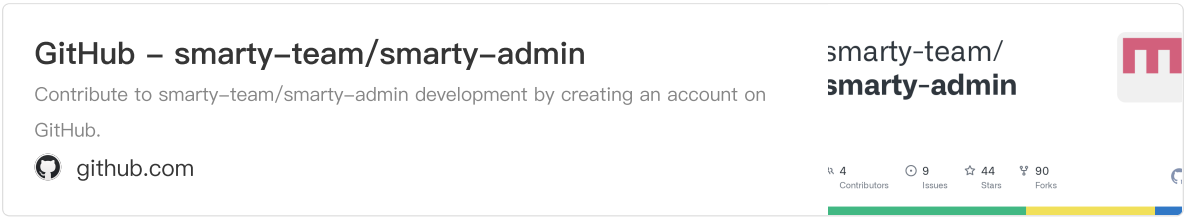
🚗 实战篇

- Monorepo 工程(Pnpm)
- 组件库工程化(Rollup)
- 中后台前端架构实战(Vite + Type...

- 手写 Vue Test Unit 框架
- 手写 Vue 测试框架
- 手写 Jest 框架

> 👩 **CLI 工具 – React 组件库 CLI**
>
> 组件库工程化 – 重置版(Rollup + TailwindCSS + VitePress)

参考代码



软件工程与工程化

软件危机

软件危机．落后的软件生产方式无法满足迅速增长的计算机软件需求，从而导致软件开发与维护过程中出现一系列严重问题的现象。

- 阿波罗登月 – 低级代码问题
- CMMI 软件成熟度模型 –

项目庞大的主要表现

- 成本不可控
- 性能与质量过低
- 项目不符合开发前需求
- 项目代码质量太低

软件工程学

作为一门新兴的学科，在应对日益庞大的规模的时候，正是缺乏工程学这样科学的方法论来帮助开发者在软件的整个生命周期中控制风险、降本提效。所以，当时人们就想将工程学的理论和方法应用到软件开发领域，从而来解决软件危机中遇到的问题。

前端工程化任务

随着用户对用户体验要求的不断提升。传统的前端编写方式已经无法满足前端的要求。

将工程方法系统化地应用到前端开发中，以系统、严谨、可量化的方法开发、运营、维护前端应用程序，目的是降本提效

开发阶段	测试阶段	构建阶段	部署阶段	监控阶段
脚手架	测试框架	编译器	持续构建	埋点平台
公共库	静态扫描工具	优化策略	验证测试	异常监控平台
包管理器	性能测试工具	部署策略		
构建工具				
调试工具				