前端工程化-Monorepo

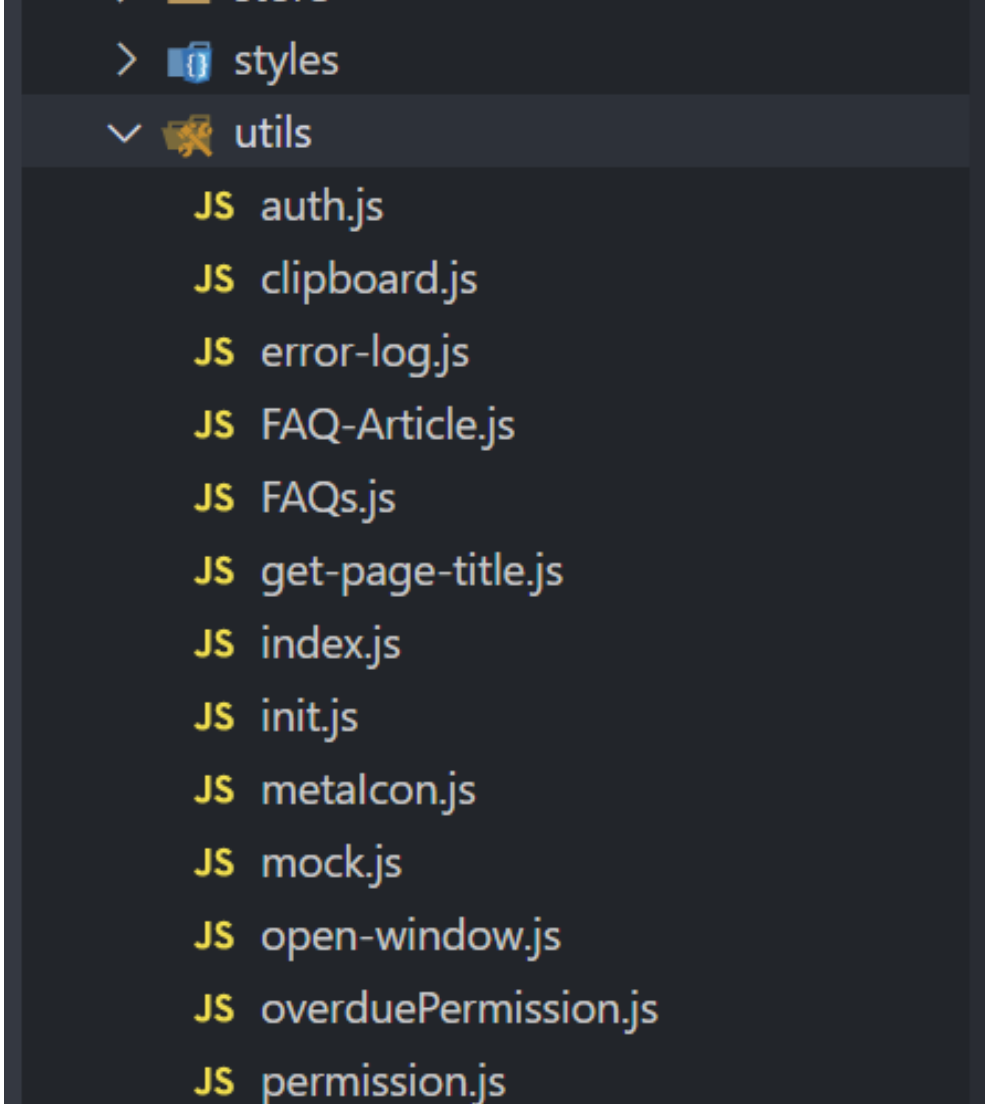
优化代码管理，提供更好的可重用性和协作性

2023/06/02

Author: huangyan321

思考：现有仓库管理方式（多仓库）会遇到什么问题？

- 工具库代码极为重复
- 依赖管理复杂，当同一个依赖在不同的仓库中使用了不同的版本时，有时可能会出现我的电脑能跑，为什么你就不能跑的疑问
- 增加团队未来增加组件库时的成本
- CI/CD的复杂性，在团队不断发展的过程中，为了方便代码管理，不可避免地需要设置和维护CI/CD流程，但多仓库模式下，每次起新项目都要在仓库中设置。



如今大部分的前端应用会采用Multi repo，即每个独立的项目都会有相对应的repo，对于维护自己的项目分清责任也很容易。（这里插入公司的各个项目）

```
Design-systems Repository
design-systems
  ├ node_modules
  ├ e2e
  ├ xxxx
Admin Repository
admin
  | ├ node_modules
  | ├ e2e
  | ├ xx...
```

这种方式看似不错，但也会带来一些问题

依赖项重复

对于我们团队的前端项目，目前选定的技术栈已经正在向 jquery -> vue2.6 -> vue2.7 -> vue3.x 的方向演进，总体技术生态已经基本固定。这时候大部分项目所安装的依赖库都有重复

// 贴上相关package

工程配置重复

贴上具体案例

- webpack、vite配置管理，兼容性问题
- 开发环境
- 多仓库手动部署

跨项目代码难共享

贴例子

Monorepo

什么是Monorepo

Monorepo 是一种项目代码管理方式，指单个仓库中管理多个项目。并不是一种工具，它只是对于项目的一种管理手段，一种思维方式。

- 统一管理 configs and tests。只有一个 repo 所以不需要再重复配置环境，包括 CI/CD、unit、e2e、webpack 都只需要维护一份就好。
- 统一管理 依赖。
- 复用模块代码以及share code
- 简化组织
- 减少重复依赖项
- 跨项目代码共享

经典开源案例

贴几个开源案例

pnpm： Workspace

贴上具体的案例

简易实现 Monorepo

项目实践

TurboRepo

Monorepo的缺点

monorepo也有一些不足的地方

- 无法管理某个、某些项目对于指定人员的权限
- 不同分支下的版本控制会显得较为混乱（medical-device bedside-terminal）
- 发布构建的难度较大
- 不适用于业务相对零散、项目之间关系不大的场景

无论是对于代码层面的设计，亦或是仓库管理层面的设计思想，都像踩跷跷板一样，没有最好，只有最适合

查阅以下链接了解更多

Turbo: what-is-a-monorepo
项目级 monorepo 策略最佳实践

Thank You