

Monorepo 工程(Pnpm)

Why should we use pnpm? pnpm 作者 Zoltan Kochan

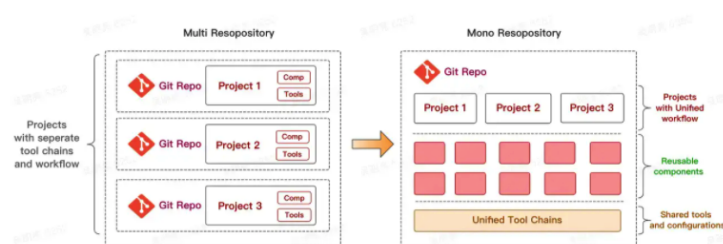
为什么现在我更推荐 pnpm 而不是 npm/yarn? 奇舞周刊

pnpm 中文官网

基于 pnpm 的 monorepo 包管理实践

<https://juejin.cn/post/7053807488952434719>

Monorepo 是什么



与 Monorepo 对比的是 MutiRepo。对于一个复杂的前端架构通常会有多个 npm package 组成。repo 指的是版本仓库。如果多个 package 放在一个 repo 仓库中就叫做 monorepo。

目前有不少大型开源项目采用了这种方式，如 `Babel`，`React`，`Meteor`，`Ember`，`Angular`，`Jest`，`Umijs`，`Vue`，还有 `create-react-app`，`react-router` 等。几乎我们熟知的仓库，都无一例外的采用了 monorepo 的方式，可以看到这些项目的第一级目录的内容以脚手架为主，主要内容都在 packages 目录中、分多个 package 进行管理。

```
├─ packages
│   └─ pkg1
│       └─ package.json
│   └─ pkg2
│       └─ package.json
└─ package.json
```

Bash

repo 其实就是 git 的仓库，monorepo

monorepo 最主要的好处是**统一的工作流**和 **Code Sharing**。比如我想看一个 package 的代码、了解某段逻辑，不需要找它的 repo，直接就在当前 repo；当某个需求要修改多个 package 时，不需要分别到各自的 repo 进行修改、测试、发版或者 npm link，直接在当前 repo 修改，统一测试、统一发版。只要搭建一套脚手架，就能管理（构建、测试、发布）多个 package。

优点

- 工作流一致性
- 项目基建成本低
- 降低依赖管理的复杂度
- 团队写作更加容易

方案

- lerna + yarn workspace
- pnpm

Pnpm 实现 Monorepo

禁用 npm

pnpm 的 monorepo 项目在 node_modules 以及开发中，项目依赖 pnpm workspace 使用 npm 或 yarn 运行时会出现问题。

package.json

```
"scripts": {  
  "preinstall": "node ./scripts/preinstall.js"  
}
```

JSON

实现当运行 npm install 或 yarn，就会发生错误并且不会继续安装。

```
if (!/pnpm/.test(process.env.npm_execpath || '')) {  
  console.log('不懂问然叔')  
  console.warn(  
    '\u001b[33mThis repository requires using pnpm as the package manager `'  
    ` for scripts to work properly.\u001b[39m\n`'  
  )  
  process.exit(1)  
}
```

Bash

建立工作空间

pnpm-workspace.yaml

```
packages:
```

TypeScript

```
# all packages in subdirs of packages/ and components/  
- 'packages/**'
```

初始化每一个 npm 包 "build:components": "node ./scripts/build.cjs"

例： ui

然后每个都执行 `npm init`，假设每个 package 的 name 依次为 `smarty-admin`
`@smarty-admin/utils`

```
mkdir ui  
pnpm init ui
```

Bash

```
├─ packages  
│   └─ admin  
│       └─ package.json # name: smarty-admin  
│       └─ utils  
│           └─ package.json # name: @smarty-admin/utils  
└─ package.json # name: @smarty-admin-monorepo
```

Bash

依赖安装

```
# 公共包安装  
pnpm i vue -w  
# 子package安装  
pnpm i vue -r --filter @smarty-admin/utils  
# 内部依赖package安装  
pnpm i @smarty-admin/utils -r --filter smarty-admin
```

Bash

依赖安装

Bash

```
# 安装在root下
pnpm i dayjs -w

# 安装在ui下
pnpm i dayjs -r --filter @smarty-admin/ui
```

版本发布与升级

pnpm+ workspace + changesets

<https://juejin.cn/post/7098609682519949325>