## yarn

### Introduction

facebook推出了yarn,用于替代npm

npm存在的一些不足：

* npm install 下载速度慢，即使是重新 install 时速度依旧慢
* 同一个项目，安装的无法保持一致性。原因是因为 package.json 文件中版本号的特点导致在安装的时候代表不同的含义。
* 使用 npm 安装多个 js 包时，包会在同一时间下载和安装。安装过程中，其中一个包抛出了一个异常，但 npm 会继续安装其他包，所以错误信息就会在一大堆提示信息中丢失掉，以至于直到执行前，都不会发现实际发生的错误。

Yarn的优点：

* 安装速度快 (服务器速度快 , 并且是并行下载)
* 版本锁定，安装版本统一
* 缓存机制，如果之前已经安装过一个软件包，用Yarn再次安装时之间从缓存中获取，就不用像npm那样再从网络下载了
* 输出简洁并且多注册来源处理。安装包时，直观地打印出必要的信息；不管包被不同的库间接关联引用多少次，只会从一个注册来源去装，防止出现混乱不一致。

### Install

#### windows

安装yarn

npm i -g yarn

设置镜像加速

yarn config set registry https://registry.npm.taobao.org

npm install -g cnpm --registry=https://registry.npm.taobao.org

检查配置：

node -v

设置为默认的远程库地址

yarn config set registry https://registry.npmjs.org

设置registry

yarn config delete registry

### Usage

#### Basic

#### 设置缓存路径

默认全局依赖包下载路径C:\Users\Administrator\AppData\Local\Yarn\Cache

yarn config set cache-folder /path/to/new/cache/folder

yarn config set cache-folder D:\yarn\_repository\cache

#### 修改 Yarn 的 data 目录

默认的data目录是C:/Users/Administrator/AppData/Local/Yarn/Data

`C:/Users/Administrator/AppData/Local/Yarn/Data/global/node\_modules` 目录是 Yarn 全局安装包的默认目录，其中包含全局安装的 Node.js 包和模块。

具体来说，以下是目录的解释：

- `C:/Users/Administrator` 是用户主目录。

- `AppData` 是 Windows 中存储应用程序数据的文件夹，通常是隐藏的。

- `Local` 子文件夹包含了与用户特定的应用程序相关的本地数据。

- `Yarn` 是 Yarn 包管理器的配置目录。

- `Data` 子文件夹存储全局数据，包括全局 Node.js 包和模块。

- `global` 子文件夹包含全局安装的 Node.js 包和模块。

在上述路径中，`node\_modules` 目录包含了通过 `yarn global add` 命令全局安装的 Node.js 包和模块。

如果您在全局安装 TypeScript 后仍然无法使用 `tsc` 命令，可能需要确保上述路径已添加到系统的 PATH 环境变量中。这样系统才能找到全局安装的 TypeScript。

使用以下命令来设置全局模块存储位置（data 目录）：

```bash

yarn config set global-folder D:\npm\_repository

```

3. 设置成功后，Yarn 将使用新路径来存储全局模块。

请注意，修改全局模块存储路径可能会影响全局安装的 Node.js 包和模块的位置。确保新路径具有足够的权限，并且不会与现有的文件夹冲突。

您还可以使用 `yarn config get global-folder` 命令来检查当前的全局模块存储路径。如果您想恢复默认的路径，可以使用 `yarn config delete global-folder` 命令删除自定义设置，Yarn 将恢复到默认的全局模块存储路径。

npm和yarn命令

|  |  |
| --- | --- |
| npm install | yarn install |
| npm install <name> --save-dev | yarn add <name> --dev |
| npm instlal --save | yarn add <name> |
| npm install -g | yarn global add |
| npm run server | yarn run server |
| npm run build | yarn run build |
| npm run start | yarn run start |