

yarn workspaces

Yarn Workspaces 是一种在多个项目中共享依赖包的强大方式，它会创建一个统一的 `node_modules` 目录，供所有工作区（项目）共享。以下是如何在多个项目中使用 Yarn Workspaces 共享 `node_modules` 的步骤：

1. **创建一个工作区根目录**：

首先，在你的文件系统中选择一个文件夹，这将是你的工作区根目录。在该目录下创建一个空的 `package.json` 文件。

2. **配置工作区**：

在工作区根目录的 `package.json` 文件中，添加一个 `workspaces` 字段，其中包含一个数组，列出你要包括在工作区的子项目目录。例如：

```
``json
{
  "private": true,
  "workspaces": [
    "projects/project1",
    "projects/project2"
  ]
}
``
```

这里的 `"projects/project1"` 和 `"projects/project2"` 是你各个子项目的相对路径。

3. **创建子项目**：

在工作区根目录下创建子项目的文件夹，例如 `projects/project1` 和 `projects/project2`。每个子项目都应该有自己的 `package.json` 文件，其中列出了项目特定的依赖。

4. **安装依赖**：

进入工作区根目录，并运行以下命令，它会安装工作区中所有子项目的依赖，并将它们放在统一的 `node_modules` 目录中：

```
``
yarn install
``
```

这会为工作区中的所有子项目安装它们各自的依赖，同时共享相同的依赖包。

5. **运行子项目**：

现在，你可以进入任何一个子项目的目录，然后运行该项目的开发服务器或构建脚本。Yarn Workspaces 会自动管理依赖的解析，确保子项目使用共享的依赖。

6. **管理依赖**：

当你需要在任何一个子项目中添加、更新或删除依赖时，只需进入该子项目的目录，运行 `yarn add`、`yarn upgrade` 或 `yarn remove` 等命令。Yarn 会自动更新工作区的 `node_modules`。

总结一下，使用 Yarn Workspaces 可以让你在多个项目中共享依赖包，同时保持项目的独立性。这种方法可以减少磁盘占用和依赖冗余，同时简化了依赖的管理和更新过程。