# 自己的Nx设计模式

对于大公司来说，有些事情要重要得多:

代码组织和命名约定  
代码所有权  
执行最佳实践  
开发人员工作流程  
部署灵活性

代码组织和命名约定

**Apps and** **Libs**

1 应用程序配置依赖注入和连接库。它们不应该包含任何组件、服务或业务逻辑。  
2 库包含服务、组件、实用程序等。它们有定义良好的公共API。

一个典型的Nx工作区拥有的库比应用程序更多，所以要特别注意libs目录的组织。

**Scope (Lib在哪里，谁拥有它)**

1 将特定于应用程序的库放到与应用程序名称匹配的目录中是一种很好的惯例。这为中小型应用程序提供了足够的组织。

happynrwl

├── apps

│ ├── happynrwlapp

│ ├── happynrwlapp-e2e

│ ├── adminapp

│ └── adminapp-e2e

├── libs

│ ├── happynrwlapp

│ │ ├── feature-main

│ │ ├── ui-table

│ │ └── utils-testing

│ ├── adminapp

│ │ ├── feature-main

│ │ ├── feature-login

│ │ ├── ui

│ │ └── utils-testing

│ └── shared

│ ├── ui

│ └── utils-testing

├── tools

├── workspace.json

├── nx.json

├── package.json

└── tsconfig.json

2 对于较大的项目，将库分组到应用程序部分是一个好主意。

happynrwl

├── apps

├── libs

│ ├── happynrwlapp

│ │ ├── registration

│ │ │ ├── feature-main （1）

│ │ │ ├── feature-login

│ │ │ ├── ui-form

│ │ │ └── utils-testing

│ │ ├── search

│ │ │ ├── feature-results

│ │ │ └── utils-testing

│ │ └── shared

│ │ └── ui (2)

│ ├── adminapp

| └── shared

│ ├── ui (3)

│ └── utils-testing

├── tools

├── workspace.json

├── nx.json

├── package.json

└── tsconfig.json

1. happynrwlapp/registration/feature-main 在一个地方使用的有作用域的库

(2) happynrwlapp/shared/ui--在单个应用程序中使用的共享库

(3) shared/ui--跨应用程序使用的共享库

**门户网站**许多企业应用程序都是门户:在运行时加载不同模块的瘦外壳。如果这是你正在构建的，以下可能是一个更好的起点:

happynrwl

├── apps

│ ├── happynrwlapp

│ ├── happynrwlapp-e2e

├── libs

│ ├── shell

│ │ └── feature-main

│ ├── registration

│ │ ├── feature-main

│ │ ├── feature-login

│ │ ├── ui-form

│ │ └── utils-testing

│ ├── search

│ │ ├── feature-results

│ │ └── utils-testing

│ └── shared

│ ├── ui

│ └── utils-testing

├── tools

├── workspace.json

├── nx.json

├── package.json

└── tsconfig.json

**Type (library 里有什么)**

使用Nx，我们可以将代码划分为具有良好定义的公共API的小型库。因此，我们可以根据所包含的内容对库进行分类。

以下是一些常见的库类型:ui, data-access, feature, util

**依赖关系管理**

对于大型组织来说，确定项目之间如何相互依赖是至关重要的。例如:

1 范围更广的库(如shared/ui)不应该依赖于范围更小的库

(如happynrwlapp/search/utils-testing)。

2 组件库应该只依赖于其他组件库和实用程序库，而不应该依赖于特性库。

Nx提供了一种称为标签的特性，可以使用它来编纂和静态地执行这些规则。点击这里阅读更多关于标签的内容。

执行最佳实践

使用Nx，我们可以通过使用工作空间图表(workspace schematics)和工作空间(workspace)连接检查来帮助团队采用最佳实践。