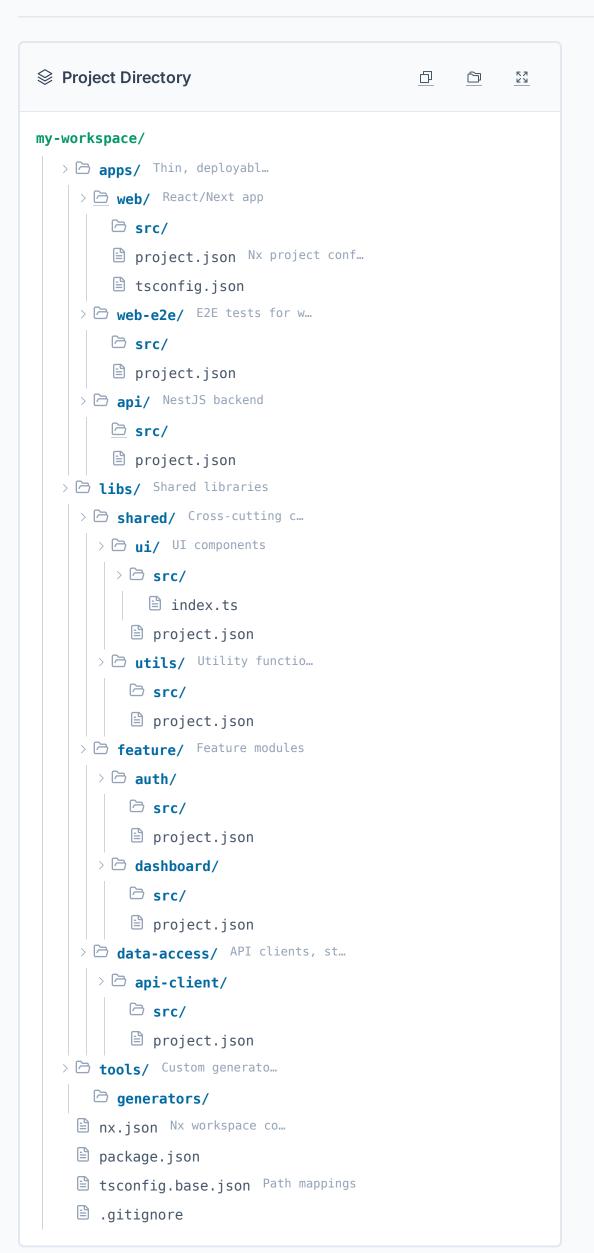


Nx with libs/apps separation. Affected commands and code generators.

#monorepo #nx #workspace #typescript #enterprise





○ Why This Structure?

Nx enforces boundaries with its libs architecture. Apps are thin shells; logic lives in libs. nx affected runs only what changed. Generators scaffold consistent code. The dependency graph powers intelligent caching and parallel execution.

apps/ - Thin shells that compose libs

libs/shared/ - Cross-app utilities and components

libs/feature/ - Domain-specific feature modules

libs/data-access/ - API clients and state management

feature - Smart components with business logic

ui - Presentational/dumb components

data-access - State, API calls, data fetching

util - Pure functions, helpers

</> Getting Started

npx create-nx-workspace@latest
nx generate @nx/react:lib shared/ui
nx affected --target=build

☑ When To Use This

- Large teams with module boundaries
- Enterprise apps with strict architecture
- Need code generators for consistency
- Complex dependency graphs

বা Trade-offs

Steeper learning curve - More concepts than Turborepo

Project-per-lib overhead - Many project.json files

Nx-specific patterns - Tighter coupling to Nx tooling