

ToiletPaper #127

Entwickeln wie Google, Facebook und Co.? Hallo Nx Workspace!

Autor: Robert Gruner / Software Engineer / Standort Leipzig

✗ Problem

Das nächste Projekt steht an. Welche Plattform soll es sein? **Nodejs**. Ok! Welche Sprache darf man nutzen? Gerne **TypeScript**! Frontend? Klar. Backend? Ja. Datenbank? Wäre gut. Hätte das gern aber alles in einem Git Repository wegen Code Sharing, Dependency Management und einfacherer Kollaboration. Kein Problem!

✓ Lösung

Schaut euch mal den "Nx Workspace" an! Das Projekt Scaffolding wird durch die CLI im Handumdrehen erledigt und ihr könnt aus verschiedenen Voreinstellungen wählen.

➔ Beispiel

```
npx create-nx-workspace@latest coffee-workspace
```

Das Kommando ist interaktiv und fragt euch, welche Technologien ihr nutzen wollt. Das sieht dann z. B. so aus:

```
? What to create in the new workspace - react-express [a workspace with a full stack application (React + Express)]
? Application name - coffee
? Default stylesheet format - styled-components [ https://styled-components.com ]
```

Was kommt dabei raus? Ein vollständig funktionierendes Projekt! Das komplette Projekt-Tooling ist vorkonfiguriert und ihr könnt die App direkt starten, bauen, testen, linten etc. Die Ordnerstruktur ist simpel und nachvollziehbar und das Ganze soll auf Best Practices aus dem Hause Google, Facebook, Microsoft und Co. basieren. Klingt etwas reißerisch, aber die Kollegen haben ganz gute Argumente an der Hand (<https://nx.dev/angular/fundamentals/monorepos-automation>).

? Fragen

Full-Stack in einem Projekt? Wird das nicht zu viel?

Nein! Man kann die einzelnen Applications einfach über deren Namen aufrufen, z.B: "nx build api-interfaces". Zudem erlaubt es die CLI, alle nötigen Kommandos auch mit Präfix "affected" laufen zu lassen. Dabei werden nur die Teile des Projektes gebaut/getestet/gelintet, welche von den derzeitigen Source-Code-Änderungen betroffen sind. Beispiel: "nx affected:test"

Wie share ich Code zwischen Frontend und Backend?

Über "Libs". Im Projekt Scaffold ist bereits eine "api-interfaces" lib enthalten, ihr könnt euch über die CLI einfach weitere generieren lassen. Beispiel: "nx generate @nrwl/workspace:library coffee-lib"

Ich mach' nur ein Proof of Concept - das ist doch bestimmt zu fancy für das kleine Wegwerf-Projekt?

Ganz im Gegenteil! Gerade am Anfang des Projektes nimmt euch die CLI viel Code-Generation ab, z.B. mit Kommandos wie "nx generate component CoffeeDashboard --project=coffee"

Überzeugt mich noch nicht. Was hat das sonst noch für Features?

"Out-of-the-box" Testing mit modernen Tools wie Jest & Cypress; Tagging von Applications & Libs im Workspace, sodass z.B. Abhängigkeiten besser abgebildet werden können ("App A baut auf Lib C auf" oder "App B darf Lib B nicht konsumieren"); Automatische Formatierung der Dateien in eurem Workspace...

+ Weiterführende Aspekte

- Getting Started: <https://nx.dev/angular/getting-started/getting-started> → Angular ist default, aber Frontend mit Web Components oder React wird auch unterstützt.
- Github: <https://github.com/nrwl/nx>
- Für API-Doku Fetischisten: <https://nx.dev/angular/api/home>