ToiletPaper #127

Developing like Google, Facebook and co.?
Hello Nx Workspace!

Author: Robert Gruner / Software Engineer / Office Leipzig

Problem

The next project is about to start. Which platform should it be? **Nodejs**. Okay! Which language can be used? **TypeScript** is appreciated! Front-end? Sure. Back-end? Yes? Database? That'd be good. But it should be in one Git repository for code sharing, dependency management and easier collaboration. No problem!

Solution

Check out the "Nx Workspace"! The project scaffolding is done instantly with the CLI and you can choose from different presets.

Example

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

The command is interactive and asks you which technologies you would like to use. It might look something like this:

? 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]

What's the outcome? A fully functioning project! The complete project tooling is pre-configured, and you can start, build, test, lint etc. the app directly. The folder structure is simple and easy to understand and the whole thing should be based on best practices from Google, Facebook, Microsoft and co. Sounds a bit lurid, but the colleagues have quite good arguments (https://nx.dev/angular/fundamentals/monorepos-automation).

Questions

Full stack in a project? Isn't that too much?

> No! You can call up the individual applications simply by their names, e.g.: "nx build api-interfaces". Furthermore, the CLI allows you to run all necessary commands with the prefix "affected". Only those parts of the project are built/tested/linted which are affected by the current source code changes. Example: "nx affected:test"

How do I share code between front-end and back-end?

> About "Libs". The project scaffolding already contains an "api-interfaces" lib, you can easily generate more via the CLI. Example: "nx generate @nrwl/workspace:library coffee-lib"

I'm just doing a proof of concept - that's probably too fancy for this little throw-away project, isn't it?

> Quite the contrary! Especially at the beginning of the project, the CLI saves you a lot of code generation, e.g. with commands like "nx generate component CoffeeDashboard --project=coffee".

I'm not convinced yet. Are there more features?

> "Out-of-the-box" testing with modern tools like Jest & Cypress; tagging of applications & libs in your workspace for better mapping of dependencies ("app A builds on lib C" or "app B must not consume lib B"); automatic formatting of files in your workspace...

Further Aspects

- Getting Started: https://nx.dev/angular/getting-started/getting-started → Angular is default, but front-end with web components or React is supported.
- Github: https://github.com/nrwl/nx
- For API documentation geeks: https://nx.dev/angular/api/home