

Practical Nx



Sensei Meetup

Tomas Horacek - 30/03/2022



- open-source toolbox for monorepos
- extensible (plugins)
- language agnostic
 - 1st class JavaScript / TypeScript support
 - but also others (.Net, Rust, ...)
- my experience:
 - 2 Next.js projects (production-ready™)
 - dev, build, test, lint, CI/CD
 - +1 "custom React" in progress

Monorepo ?

<https://monorepo.tools/>

Plan for Today

- migrate simple React/Express.js app to Nx
- break React monolith to smaller libraries

Quacker App



- "copy" of Twitter
- example for students at VŠE university
- very simple - good structure
- it's overkill
- use your imagination 🧘



Demo

- parts:
 - design system
 - auth
 - navigation
 - quack

Backend

- Express.js
- GraphQL - Apollo Server
- TypeScript (not really good 😅)
- **mocked data** (or MySQL)

Frontend

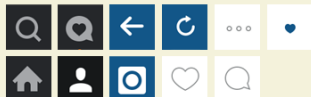
- React + Create React App
- GraphQL - Apollo Client
- **Tachyons CSS** (utility based - like Tailwind)
- TypeScript

Frontend - Atomic Design

<https://atomicdesign.bradfrost.com>



ATOMS



PHOTO

XXXXXX likes

thisistheusersinstagramhandle

2h

Lorem ipsum dolor sit amet, consectetur adipiscing

150 x 150

1080 x 1080



MOLECULES



thisistheusersinstagramhandle 2h

1080 x 1080



XXXXXX likes

thisistheusersinstagramhandle Lorem ipsum dolor sit amet, consectetur adipiscing



ORGANISMS



thisistheusersinstagramhandle 2h

1080 x 1080



XXXXXX likes
thisistheusersinstagramhandle Lorem ipsum dolor sit amet, consectetur adipiscing



TEMPLATES



thisistheusersinstagramhandle 2h

1080 x 1080



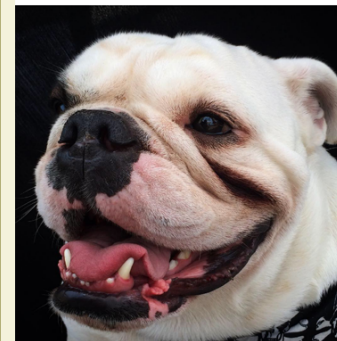
XXXXXX likes
thisistheusersinstagramhandle Lorem ipsum dolor sit amet, consectetur adipiscing



PAGES



brad_frost 2h



41 likes
brad_frost Hi!








Initiate Workspace

```
cd ..  
npx create-nx-workspace@latest --packageManager=yarn  
# and move files to Quacker repo
```

- select:
 - **quacker** - Workspace name
 - **apps** - What to create in the new workspace
 - **No** - Use Nx Cloud

Steps

- add Nx plugins
- generate empty apps (backend - frontend)
- move backend to Nx  BACK
- move frontend to Nx 
- break frontend to libraries   

Add Nx Plugins

```
yarn add --dev @nrwl/react @nrwl/express @nrwl/node
```

- add Nx **React Plugin** and **Express Plugin**

Generate Express.js App

```
yarn nx list  
yarn nx list @nrwl/express  
yarn nx g @nrwl/express:app --help
```

```
yarn nx g @nrwl/express:app quacker-backend --dryRun  
yarn nx g @nrwl/express:app quacker-backend
```

- generate Express.js app

Generate Frontend App

```
yarn nx list @nrwl/react  
yarn nx g @nrwl/react:app --help
```

```
yarn nx g @nrwl/react:app quacker-frontend \  
  --style=css --routing=false --pascalCaseFiles --dryRun
```

```
yarn nx g @nrwl/react:app quacker-frontend \  
  --style=css --routing=false --pascalCaseFiles
```

Run Dev Servers

```
yarn nx run quacker-backend:serve
```

```
yarn nx run quacker-frontend:serve
```

```
yarn nx run-many --target=serve \  
  --projects=quacker-backend,quacker-frontend --parallel
```


Backend Packages

```
cd backend
```

```
jq -r '.dependencies | [ keys[] as $k | "\($k)@\(.[$k])" ] | join("\n")' package.json
```

```
cd ..  
yarn add apollo-server-core@3.6.6 apollo-server-express@3.6.6 \  
argon2@0.28.5 body-parser@1.19.2 cors@2.8.5 dotenv-flow@3.2.0 \  
graphql@15.8.0 jsonwebtoken@8.5.1 mariadb@2.5.6
```

<https://stedolan.github.io/jq/>

Move Backend



```
cd backend/src
git mv * ../../apps/quacker-backend/src/
cd ..
git mv db .env .gitignore README.md ../apps/quacker-backend

cd ../apps/quacker-backend/src
git rm main.ts
git mv index.ts main.ts
cd ../../..
```

```
// Add to apps/quacker-backend/tsconfig.json
{
  //...
  "compilerOptions": {
    "esModuleInterop": true
  },
  //...
}
```

Frontend Packages



```
cd frontend
```

```
jq -r '.dependencies | [ keys[] as $k | "\($k)@\(.[$k])" ] | join("\n")' package.json
```

```
cd ..
```

```
yarn add @apollo/client@^3.5.10 \  
@fortawesome/fontawesome-svg-core@6.1.1 \  
@fortawesome/free-regular-svg-icons@6.1.1 \  
@fortawesome/free-solid-svg-icons@6.1.1 \  
@fortawesome/react-fontawesome@0.1.18 \  
@testing-library/react@^12.1.4 \  
@testing-library/user-event@^13.5.0 \  
@types/react-router-dom@^5.3.3 classnames@2.3.1 \  
date-fns@2.28.0 formik@2.2.9 graphql@^15.8.0 lodash@4.17.21 \  
react@^17.0.2 react-dom@^17.0.2 \  
react-router-dom@5.3.0 tachyons@4.12.0 yup@0.32.11
```

Move Frontend

```
git rm -rf apps/quacker-frontend/src/app
```

```
cd frontend
```

```
git mv -f ./src/index.tsx \
```

```
  ../apps/quacker-frontend/src/main.tsx
```

```
git mv -f ./src/index.css \
```

```
  ../apps/quacker-frontend/src/styles.css
```

```
git mv -f ./public/favicon.ico \
```

```
  ../apps/quacker-frontend/src/favicon.ico
```

```
git mv ./src ../apps/quacker-frontend/src/app
```

```
git mv -f ../.env ../README.md ../apps/quacker-frontend
```

```
cd ..
```

Fixes in: apps/quacker-frontend #1

```
// <root>/tsconfig.base.json
{
  "compilerOptions": {
    // ...
    "paths": {
      "src/*": [ "apps/quacker-frontend/src/app/*" ]
    }
  },
  // ...
}
```

```
// tsconfig.json
{
  "compilerOptions": {
    // ...
    "noPropertyAccessFromIndexSignature": false,
    // ...
  }
}
```

Fixes in: apps/quacker-frontend #2

```
// tsconfig.app.json - add to `exclude`  
// tsconfig.spec.json - add to `include`  
"**/_ _tests_ _/**/*.ts",  
"**/_ _tests_ _/**/*.tsx",
```

```
// src/main.tsx - remove  
import 'tachyons';  
import './index.css';  
  
// src/styles.css - add  
@import 'tachyons/css/tachyons.css';
```

```
// src/index.html  
<title>Quacker</title>  
<body class="w-100 sans-serif black-90 bg-white"></body>
```

Fixes in: apps/quacker-frontend #3

```
yarn nx run quacker-frontend:lint
```

```
// .eslintrc.json
"files": ["*.ts", "*.tsx", "*.js", "*.jsx"],
// ...
"rules": {
  "@nrwl/nx/enforce-module-boundaries": "off"
}

// <root>/.eslint.json
"files": ["*.ts", "*.tsx"],
//...
"rules": {
  "@typescript-eslint/no-empty-function": "off",
  "@typescript-eslint/ban-types": "off"
}
```

ENV Variables

- Nx docs
- REACT_APP_ -> NX_

Fixes

```
yarn nx format --all  
yarn nx run-many --target=lint --all --fix
```

Dependency Graph & Affected

```
yarn nx graph --open=false --watch
```






```
yarn nx format
```

```
yarn nx affected --target=format
```

```
yarn nx affected --target=lint
```

```
yarn nx affected --target=test
```

Steps

- ☒ add Nx plugins
- ☒ generate empty apps (backend + frontend)
- ☒ move backend to Nx  BACK
- ☒ move frontend to Nx 
- ☐ break frontend to libraries   

Design System Library

```
yarn nx g @nrwl/react:lib --help
```

```
yarn nx g @nrwl/react:lib ui \  
--directory=shared/design-system --style=none \  
--importPath=@quacker/ui
```

```
git mv apps/quacker-frontend/src/app/atoms \  
libs/shared/design-system/ui/src/lib
```

```
# Move molecules/{Field.tsx, FormikField.tsx,  
# LoadingButton.tsx, ReloadButton.tsx} to design-system
```

```
yarn nx affected:lint
```

Navigation and Auth Libraries

```
# import { route } from 'src/Routes'  
yarn nx g @nrwl/react:lib utility \  
  --directory=navigation --style=none
```

```
# src/utils/auth  
yarn nx g @nrwl/react:lib context-ui \  
  --directory=auth --style=none
```

```
# TopNavigation  
yarn nx g @nrwl/react:lib ui \  
  --directory=navigation --style=none
```

```
# SignIn/Up forms and templates  
yarn nx g @nrwl/react:lib ui \  
  --directory=auth --style=none
```

```
# SignIn/Up pages  
yarn nx g @nrwl/react:lib feature \  
  --directory=auth --style=none
```

Quack Libraries

```
yarn nx g @nrwl/react:lib ui \  
  --directory=quack --style=none
```

```
yarn nx g @nrwl/react:lib feature \  
  --directory=quack --style=none
```

Nx Affected

```
yarn nx affected:graph --open=false
```





```
# CI/CD
```

```
yarn --silent nx affected:apps --plain
```

```
yarn --silent nx affected:libs --plain
```

Recapitulation



- initiate Nx tools 
- Nx plugins 
- generate apps and libs
- tasks: serve, lint, format
- dependancy graph 
- affected 

More Resources



- <https://monorepo.tools/>
- <https://nx.dev/>
- Effective React Development with Nx - free 90 pages PDF

Thank you for your attention! 🙏

<https://github.com/heracek/practical-nx>

- `initial-state` branch
- `main` branch (state at the end of presentation)
- `final` branch (full monorepo)