

# Cześć



Kacper Grzeszczyk

[kacgrzes.io](http://kacgrzes.io) / [kacgrzes@gmail.com](mailto:kacgrzes@gmail.com)

# Jestem programistą JS / TS :



- Mobile (React Native) - 60%
- Web (React) - 30%
- Server (NodeJS) - 10%

# Byłem trenerem w InfoShare Academy



# Budowanie wieloplatformowych aplikacji za pomocą narzędzi NX



andre bass  
@onlyandrebass

Replying to @elonmusk

Twitter: "We will give you a seat on our board."

Elon Musk:



4:14 PM · Apr 14, 2022 · Twitter for iPhone

**zbudujemy klon Tittera roboczo  
nazwany Tweeter**



# Disclaimer

“ Budowanie rozbudowanego klonu na potrzeby prezentacji mija się z celem. Do aplikacji z całą pewnością da się dorzucić więcej funkcjonalności, ale celem prezki jest przede wszystkim pokazanie możliwości budowania projektów na platformie NX.

”

# Jak podejść do budowania wieloplatformowej aplikacji?

# <ScianaTekstu>

# Sposób #1 - Polyrepo

“ Każdy projekt / aplikacja ma swoje własne repozytorium. W wielu projektach w których pracowałem, to obecnie standard pisania aplikacji. Tworzy się repozytoria z monolitami. api, admin, mobile, web, każdy projekt ma swoje własne repo. Jak są jakieś części wspólne to wyciąga się je do osobnych repozytoriów.

”

## Sposób #2 – Monorepo

“ Cały kod projektu jest trzymany w jedym repozytorium z wieloma bibliotekami / aplikacjami / projektami. ”

# Monolit

“ Jeśli repozytorium zawiera masywną aplikację bez podziału i hermetyzacji poszczególnych części, jest to po prostu duże repo. Nie oznacza to jednak wcale, że jest to monorepo. W rzeczywistości, takie repo jest nadmiernie monolityczne, co często jest pierwszą rzeczą, jaka przychodzi ludziom do głowy, gdy myślą o monorepo. ”

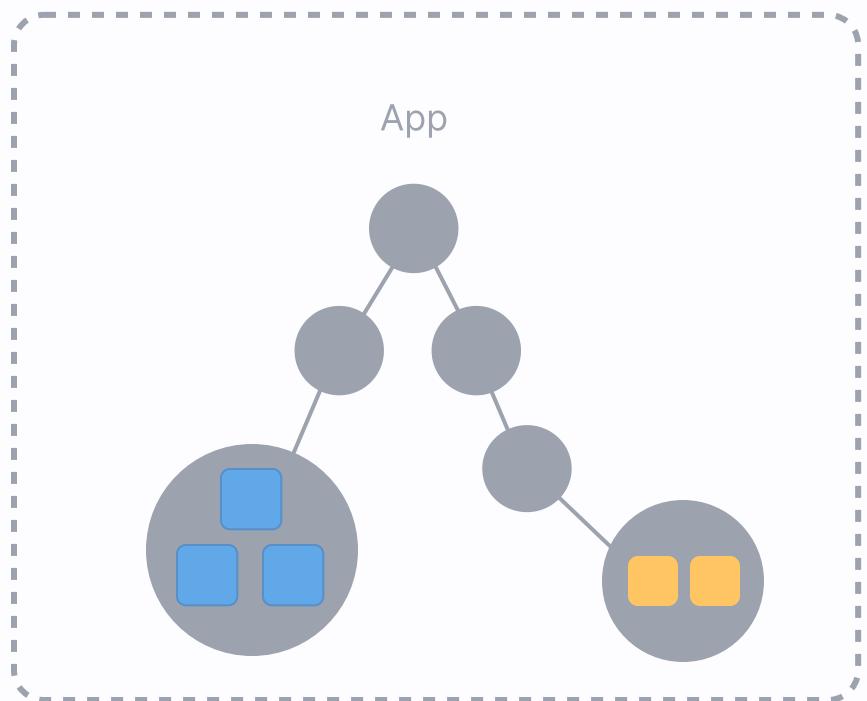
Monolit != Monorepo

# Moduł

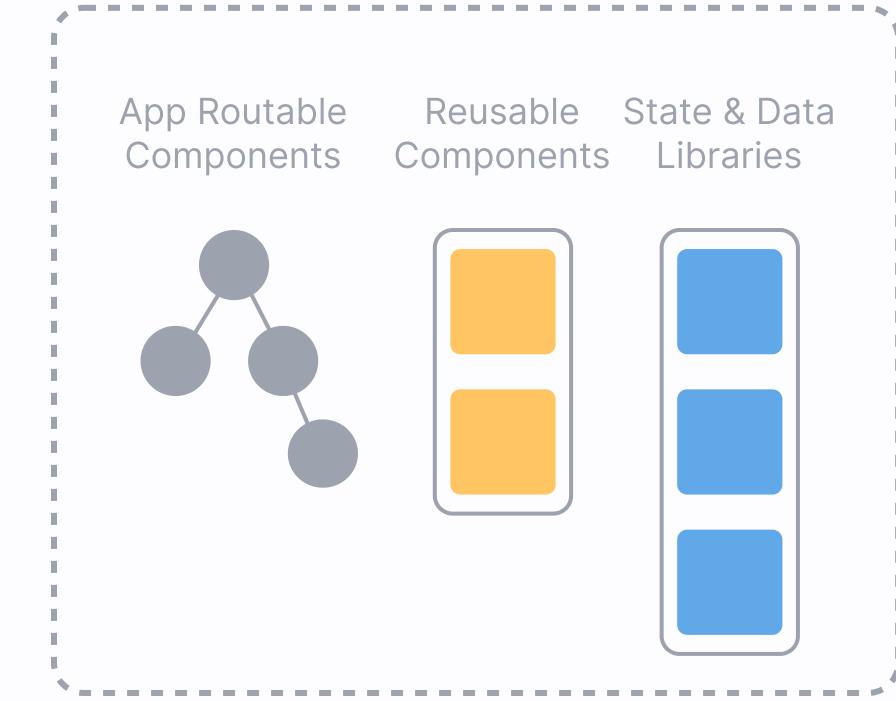
“ oddzielny twór, przeważnie w postaci osobnego pliku, zawierający zdefiniowany interfejs, a także implementacje typów wartości, klas, zmiennych, stałych oraz treści procedur i funkcji. Jest to podstawowy element koncepcji programowania modularnego pozwalający na podział kodu programu na funkcjonalne części i umieszczenie ich w osobnych modułach, które są ponadto niezależne i wymienne.

”

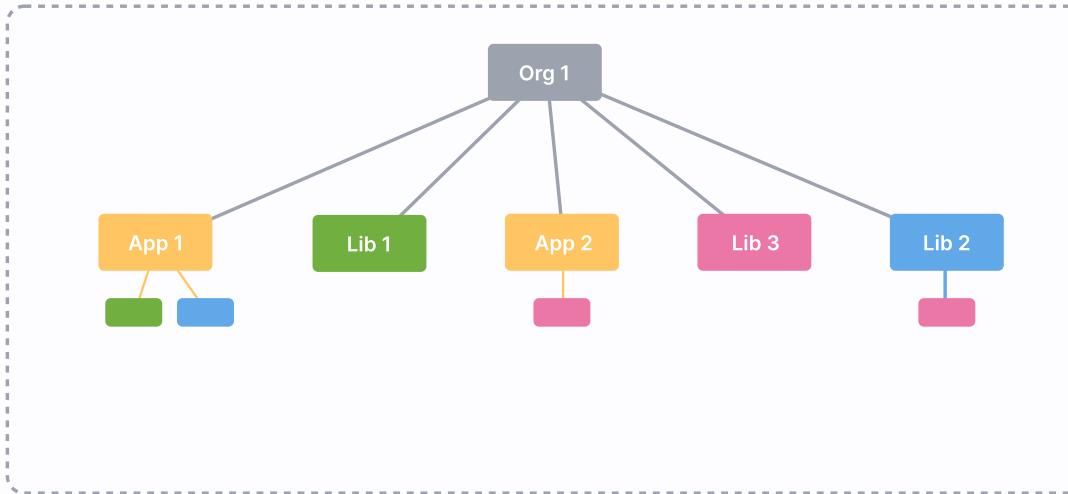
## Monolith App Repo



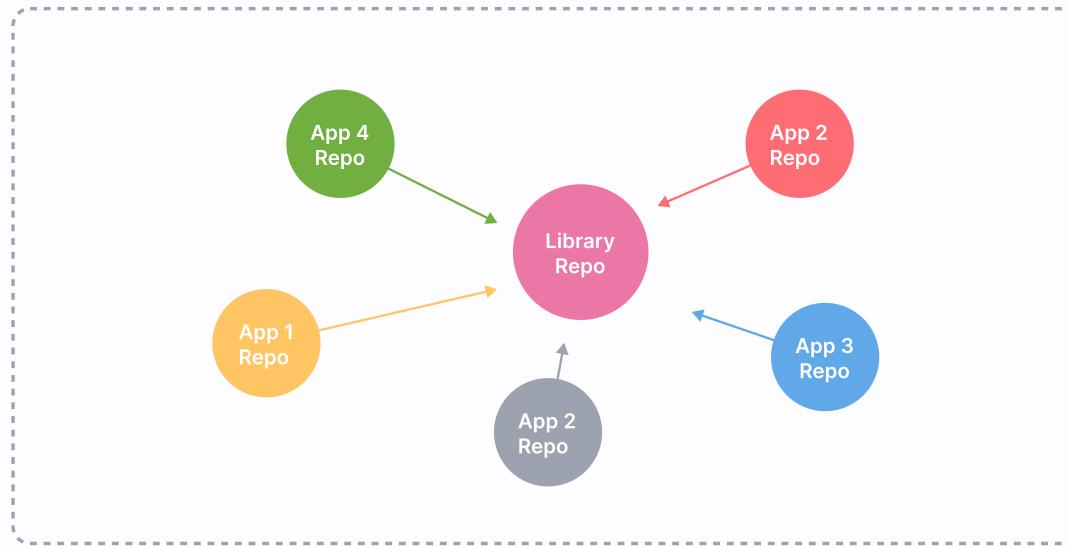
## Modular App Repo



## Monorepo



## Polyrepo



 więcej na:  
<https://monorepo.tools/>

</ScianaTekstu>

# NX Dev

<https://nx.dev>

“ NX jest jak VSCode wśród narzędzi do budowania. Core Nx'a umożliwia tworzenie i analizę grafu projektu, orkiestrację, wykonywanie zadań oraz generowanie kodu. ”

Przyjrzyjmy się temu z lotu ptaka 

# Tworzenie kodu

```
nx generate [generator] [project]  
  
nx generate @nrwl/react:app my-app  
nx generate @nrwl/react:lib shared-button  
nx generate @nrwl/react:storybook-configuration shared-button  
nx generate @nrwl/js:lib my-lib
```

Nx rozwiązuje problem konfiguracji narzędzi takich jak: Jest, Babel, TypeScript, Webpack, Prettier, Eslint

# Działanie na kodzie

```
nx [target] [project]
nx build app
nx serve app
nx test app
```

Dzięki Nx każdy projekt ma te same polecenia do budowania, testowania i uruchamiania.

# Rozumienie kodu

```
nx graph  
nx list
```

Nx analizuje strukturę naszego kodu co umożliwia podgląd jakie moduły są od siebie zależne, jak ze sobą współpracują. Wykrywa zmiany w kodzie co pozwala wykonywać polecenia tylko na zmodyfikowanych projektach.

**Zapamiętajmy i wróćmy do  
aplikacji...**

**Określamy jakie fizyczne sa do zaimplementowania**

# **W aplikacjach mobilnych (ios oraz android)**

## **i w apce webowej :**

- przeglądanie tweetów
- dodawanie tweetow
- dodawanie komentarzy
- like / unlike tweeta

# W aplikacji adminowej 🖥 (next.js)

- usuwanie userów i tweetów 😱

# **W aplikacji serwerowej**

endpointy do wcześniej wymienionych funkcjonalności:

- auth - logowanie
- users - rejestracja, listowanie oraz usuwanie użytkowników
- tweets - dodawanie, listowanie oraz usuwanie tweetów oraz like / unlike tweeta
- comments - listowanie, dodawanie, usuwanie komentarzy

# Rozpoczęcie projektu

```
› npx create-nx-workspace@latest
Need to install the following packages:
create-nx-workspace@latest
Ok to proceed? (y) y
✓ Workspace name (e.g., org name)      · infoshare-f3
✓ What to create in the new workspace · apps
✓ Use Nx Cloud? (It's free and doesn't require registration.) · No
```

> NX Nx is creating your v14.0.3 workspace.

To make sure the command works reliably *in* all environments, and that the preset is applied correctly, Nx will run "npm install" several times. Please wait.

- ✓ Installing dependencies with npm
- ✓ Nx has successfully created the workspace.

```
.  
|   └── apps  
|   └── dist  
|   └── libs  
|   └── node_modules  
|   └── presentation  
|   └── tools
```

nie miałem projektu UI, więc na samym początku zacząłem od robienia aplikacji mobilnej w React Native

```
npm install @nrwl/react-native --save-dev  
nx g @nrwl/react-native:app tweeter-mobile
```

Polecenie tworzy dwie aplikacje w katalogu `apps/`

- `tweeter-mobile`
- `tweeter-mobile-e2e`

# NX Console

Można używać wtyczki do VSCode, która pozwala zapomnieć o poleceniach i flagach, których używalibyśmy z poziomu CLI

Generate — nx-examples

NX CONSOLE ...

GENERATE & RUN TARGET

- Generate
- Run
- Build
- Serve
- Test
- ...

COMMON NX COMMANDS

- dep-graph
- run-many
- affected
- affected:apps
- affected:build
- affected:dep-graph
- affected:e2e
- affected:libs
- affected:lint
- affected:test

PROJECTS

- > cart
- > cart-cart-page
- > cart-e2e
- > products
- > products-e2e
- > products-home-page
- > products-product-detail-page
- > shared-assets

Nx Generate X

Search flags

Run

Name \*

Project \*

Style

Directory

Class Component

Export

Flat

Global Css

Js

Pascal Case Directory

Pascal Case Files

Routing

Skip Tests

**nx generate @nrwl/react:component** 

**name \***  
The name of the component.

**project \***  
The name of the project.

**style**  
The file extension to be used for style files.

**css** 

**directory**  
Create the component under this directory (can be nested).

**classComponent**  
 Use class components instead of functional component.

**export**  
 When true, the component is exported from the

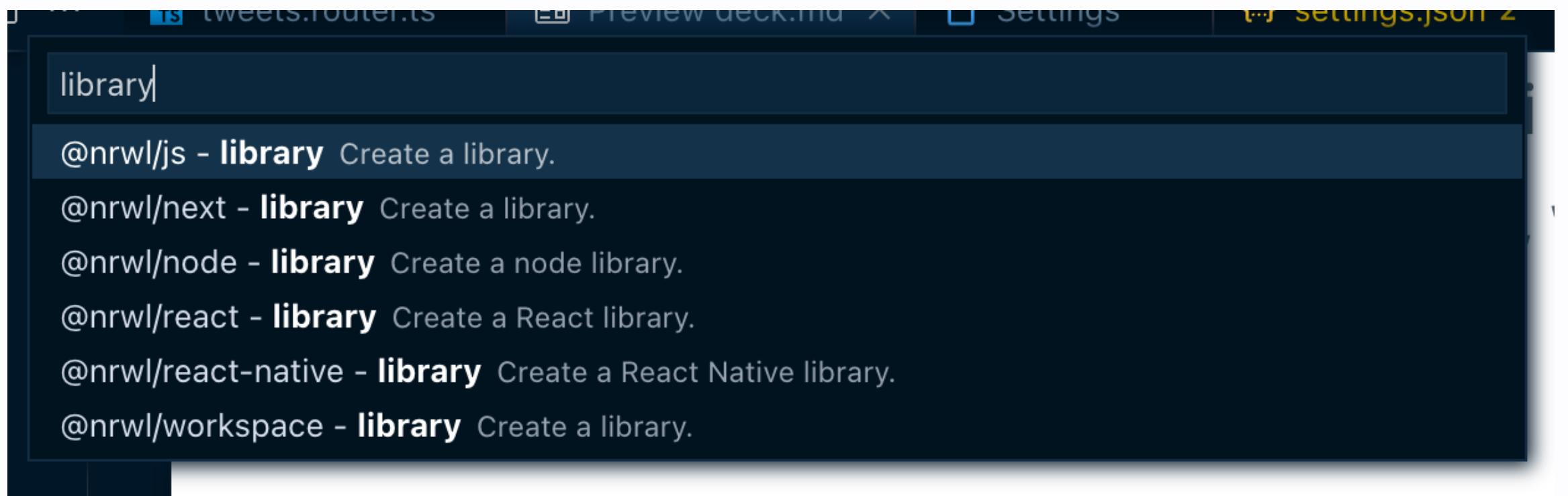
infoshare  binarapps 

34

W kolejnym kroku zrobiłem paczkę z testowymi danymi z których będę korzystał do wypełnienia UI aplikacji.

A screenshot of the Visual Studio Code interface. The top bar shows tabs for 'tweets.router.ts' (with a TS icon), 'Preview deck.md', 'Settings', and 'settings.json 2'. A search dropdown is open, displaying suggestions starting with '>nx gener'. The suggestions list includes:

- Nx Console: Focus on **Generate** & Run Target View
- Nx: generate**
- Nx: generate (ui)** (highlighted with a gear icon)
- Nx: generate application
- Nx: generate library



Search flags

Run

Name \*

Directory

Buildable

Compiler

Config

Import Path

Js

Linter

Pascal Case Files

Publishable

Set Parser Options Project

Skip Format

Skip Ts Config

Skip Type Check

Strict

Tags

Test Environment

Unit Test Runner

## nx generate @nrwl/js:library

**name** \*

Library name.

test-data

**test-data**

--directory=shared

**directory**

A directory where the lib is placed.

shared

**buildable**

Generate a buildable library.

**compiler**

The compiler used by the build and test targets

tsc

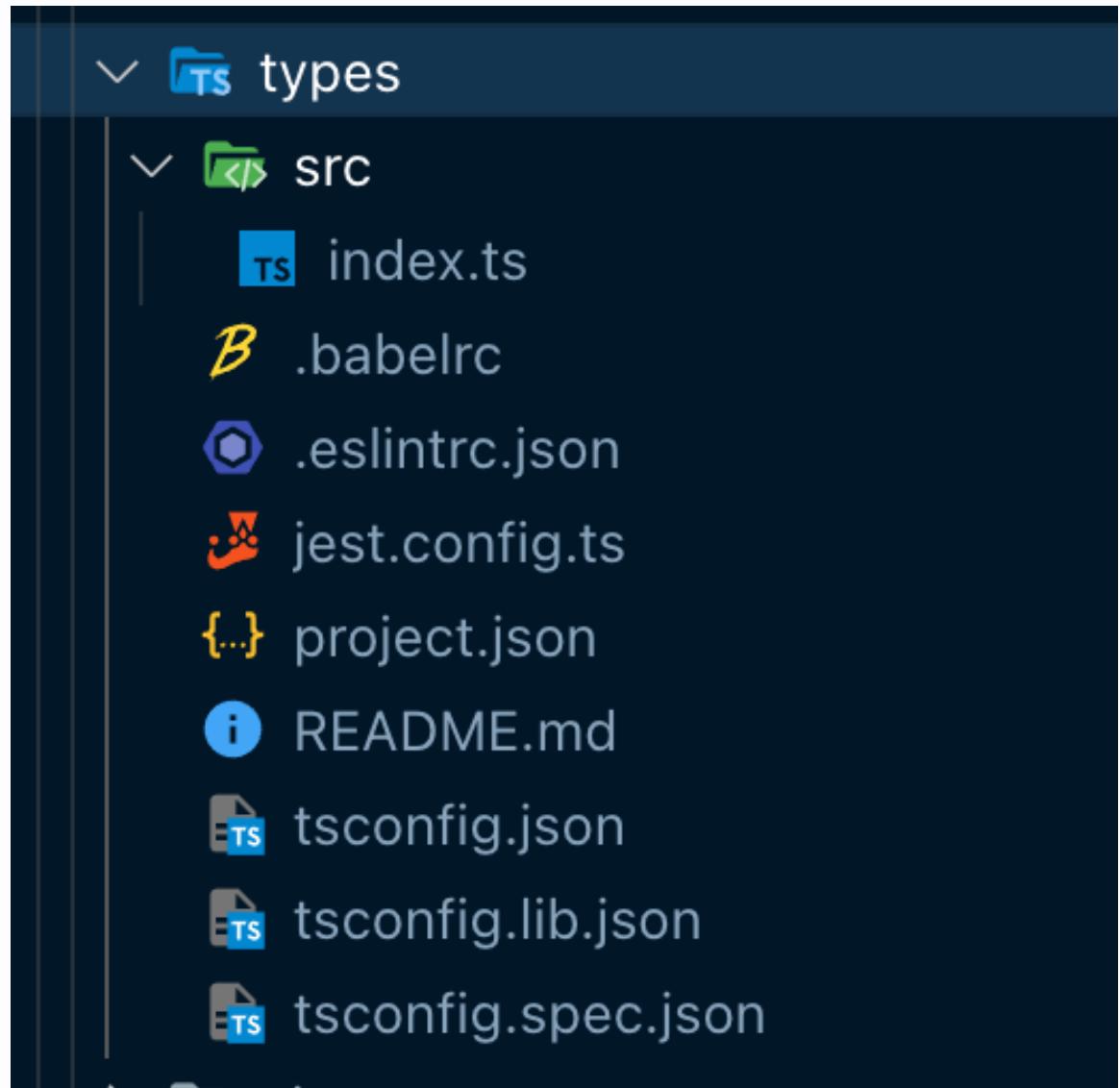


**config**

Determines whether the project's executors should be configured in

**Ale, dane testowe będą miały określony typ**

Więc generujemy kolejna paczke z typami :)



```
export type Tweet = {  
    id: string;  
    createdAt: Date;  
    text: string;  
    authorId: string;  
    replyCount: number;  
    likedCount: number;  
    liked?: boolean;  
};
```

A następnie w paczce @infoshare-f3/test-data ...

```
import { User, Role, Tweet, Comment } from '@infoshare-f3/types';
import faker from '@faker-js/faker';
import { generateRandomId } from './generateRandomId';

const createRandomTweet = (authorId: string): Tweet => {
  return {
    id: generateRandomId(),
    createdAt: faker.date.recent(),
    likedCount: 0,
    replyCount: 0,
    text: faker.lorem.words(30).slice(0, 160),
    authorId,
    liked: false
  };
};
```

Importujemy typy i używamy do funkcji generujacej tweety

a następnie generujemy testowe tweety i eksportujemy z paczki

```
export const tweets = new Array(50)
  .fill(null)
  .map(() => createRandomTweet(faker.random.arrayElement(users).id));
```

# Współdzielenie komponentów

Zastanowiłem się jak można współdzielić komponenty i dla samego faktu pokazania, że można je współdzielić między projektami, uznałem, że stworzę je przy użyciu paczek:

react-native-web oraz tailwind

czyli kolejna paczka..

Name \*  
Style  
Directory  
App Project  
Buildable  
Compiler  
Component  
Global Css  
Import Path  
Js  
Linter  
Pascal Case Files  
Publishable  
Routing  
Set Parser Options Project  
Skip Format  
Skip Ts Config  
Standalone Config  
Strict  
Tags  
Unit Test Runner

## nx generate @nrwl/react:library

**name \***

Library name

ui

ui  
--directory=shared

**style**

The file extension to be used for style files.

none



**directory**

A directory where the lib is placed.

shared

**appProject**

The application project to add the library route to.

**buildable**

Generate a buildable library.

Search flags

Name \*

Style

Directory

App Project

Buildable

Compiler

Component

Global Css

Import Path

Js

Linter

**nx generate @nrwl/react:library** 

css

scss

styl

less

styled-components

**@emotion/styled**

styled-jsx

✓ none

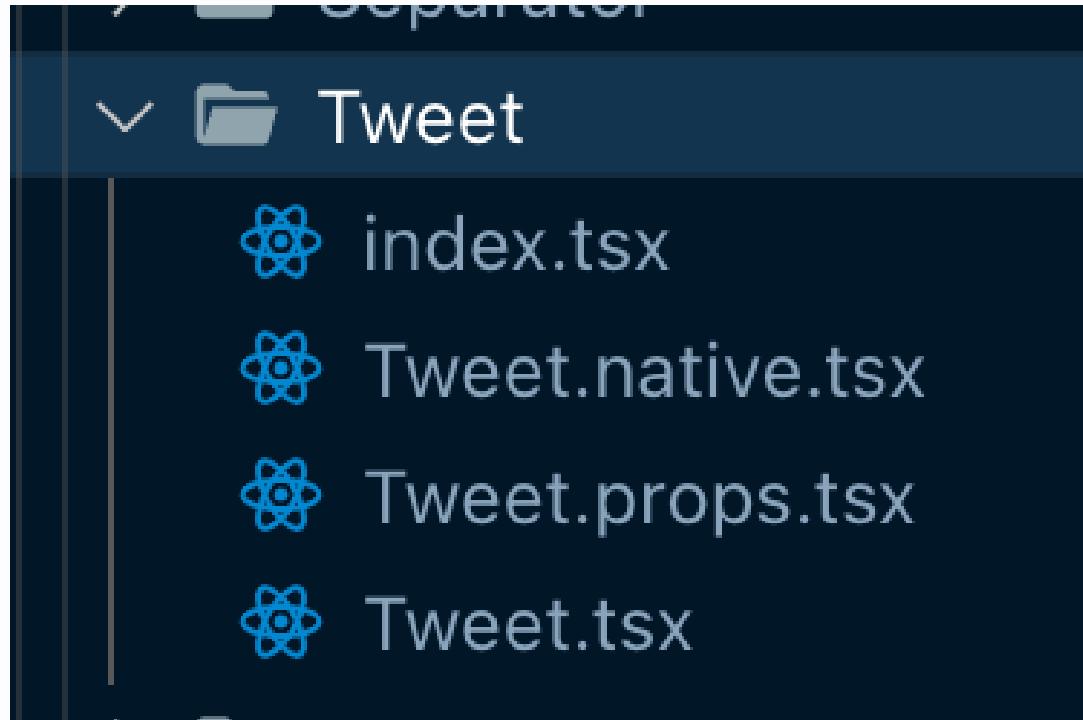
directory

```
import React, { FC } from 'react';
import { Pressable, Text } from 'react-native';
import tw from 'twrnc';

type ButtonProps = {
  size?: 'small' | 'normal';
  title: string;
  onPress?: () => void;
  disabled?: boolean;
};
```

```
export const Button: FC<ButtonProps> = ({  
  title,  
  size = 'normal',  
  onPress,  
  disabled,  
) => {  
  return (  
    <Pressable  
      disabled={disabled}  
      style={[  
        tw`  
          bg-blue-400 justify-center items-center rounded-full px-4`  
        ,  
        size === 'small' && tw`h-8 max-h-8`,  
        size === 'normal' && tw`h-11 max-h-11`,  
        disabled && tw`bg-slate-300`,  
      ]}  
      onPress={onPress}  
    >  
      <Text style={tw`text-white`}>{title}</Text>  
    </Pressable>  
  );  
};
```

**Co jesli na webie ma to wygladac  
inaczej?**



```
import { Tweet, User } from '@infoshare-f3/types';

export type TweetProps = {
    author: User;
    tweet: Tweet;
    onTweetPress?: () => void;
    onLikePress?: () => void;
    onCommentPress?: () => void;
    onAvatarPress?: () => void;
};
```

```
export const Tweet = (props: TweetProps) => {
  const {
    author,
    tweet,
    onTweetPress,
    onCommentPress,
    onLikePress,
    onAvatarPress,
  } = props;

  return (
    <div className="flex flex-col p-4 bg-white">
      <div className="flex w-full flex-row" onClick={onTweetPress}>
        <Avatar
          onPress={onAvatarPress}
          profileImageUrl={author?.profileImageUrl}
        />
```

```
export const Tweet = (props: TweetProps) => {
  const {
    author,
    tweet,
    onTweetPress,
    onCommentPress,
    onLikePress,
    onAvatarPress,
  } = props;

  return (
    <View style={tw`flex-col p-4 bg-white`}>
      <Pressable style={tw`w-full flex-row`} onPress={onTweetPress}>
        <Avatar
          onPress={onAvatarPress}
          profileImageUrl={author?.profileImageUrl}
        />
```

Zbudowałem zestaw gotowych komponentów i wyeksportowałem je z paczki

```
export * from './Avatar';
export * from './Button';
export * from './Tweet';
export * from './TweetButton';
export * from './CommentCard';
export * from './Separator';
export * from './Container';
export * from './UserCard';
```

**a następnie użyłem ich do budowania  
ekranów aplikacji**

```
import { Tweet, TweetButton, Separator } from '@infoshare-f3/ui';
import { tweets } from '@infoshare-f3/test-data';

// ...

const renderItem = useCallback(
  ({ item }) => {
  const author = item.author;
  return (
    <Tweet
      author={author}
      tweet={item}
    />
  );
},
[])

// ...

<FlatList
  data={tweets}
  renderItem={renderItem}
/>
```

Tworzyłem testowe dane dla wszystkich domen,  
podpinałem kolejne komponenty, tworzyłem ekrany i w  
rezultacie wyszło:

5:07



## Tweety



@Wellington.Lesch99 · today at 4:07 PM  
aspernatur dolores ut qui ipsam non ut  
dolore dignissimos iusto dolore est  
veritatis voluptates et velit atque  
molestiae in quidem numquam nesciunt  
qui reiciend

7

33



@Charlene.Ritchie94 · today at 3:56 PM  
excepturi expedita sequi in est quo  
eaque velit ea iusto est laboriosam quo  
voluptas ullam reprehenderit quis ut  
cumque non ad cumque sint quo quia  
eum et place

11

34



@Nick.Cruickshank · today at 2:59 PM  
quia qui rerum aut et aliquam et atque  
consequuntur ex ducimus magni quos  
non nulla quis et quidem magni  
consequatur accusamus quae dolorum  
soluta et et et sit

7

28



@Yasmin.Ward · today at 2:20 PM  
sapiente quia eum repudiandae  
consequatur dolores dicta atque id  
esse maiores nam fuga amet quae sit



5:07



## Tweet



@Wellington.Lesch99 · today at 4:07 PM  
aspernatur dolores ut qui ipsam non ut  
dolore dignissimos iusto dolore est  
veritatis voluptates et velit atque  
molestiae in quidem numquam nesciunt  
qui reiciend

7

33



animi consequatur molestias vel quia  
dolore neque voluptatem fugiat expedita  
voluptates consequatur maxime est nihil  
suscipit preferendis quasi asperiores  
deser



qui modi qui magni incident distinctio  
quod deserunt magni preferendis  
architecto qui eveniet sapiente voluptate  
facilis similique tenetur est esse  
explicabo si



quod hic sed harum vel eveniet totam  
facilis maiores ducimus optio quae ut  
quis corrupti et neque nihil neque ea  
nihil illum illo voluptatem iure  
consectetur au



ex odio et consequatur corporis  
inventore sit alias molestiae in minus  
asperiores qui rerum laboriosam qui ea

5:08



Tweet

Co się dzieje?

**Android nie wygląda już tak dobrze, ale uruchomiłem go tylko raz :) Więc jest i tak całkiem nieźle:**



@Wellington.Lesch99 · today at 4:07 PM  
aspernatur dolores ut qui ipsam non  
ut dolore dignissimos iusto dolore  
est veritatis voluptates et velit atque  
molestiae in quidem numquam nesciunt  
qui reiciend

7

33



@Charlene.Ritchie94 · today at 3:56 PM  
excepturi expedita sequi in est quo eaque  
velit ea iusto est laboriosam quo voluptas  
ullam reprehenderit quis ut cumque non  
ad cumque sint quo quia eum et place

11

34



@Nick.Cruickshank · today at 2:59 PM  
quia qui rerum aut et aliquam et atque  
consequuntur ex ducimus magni  
quos non nulla quis et quidem magni  
consequatur accusamus quae dolorum  
soluta et et et sit

7

28



@Yasmin.Ward · today at 2:20 PM  
sapiente quia eum repudiandae  
consequatur dolores dicta atque id est  
esse maiores nam fuga amet quae  
sit aspernatur et qui eum unde dolor  
accusantium architect



@Wellington.Lesch99 · today at 4:07 PM  
aspernatur dolores ut qui ipsam non  
ut dolore dignissimos iusto dolore  
est veritatis voluptates et velit atque  
molestiae in quidem numquam nesciunt  
qui reiciend

7

33



animi consequatur molestias vel quia  
dolore neque voluptatem fugiat expedita  
voluptates consequatur maxime est nihil  
suscipit perferendis quasi asperiores  
deser



qui modi qui magni incident distinctio  
quod deserunt magni perferendis  
architecto qui eveniet sapiente voluptate  
facilis similique tenetur est esse  
explicabo si



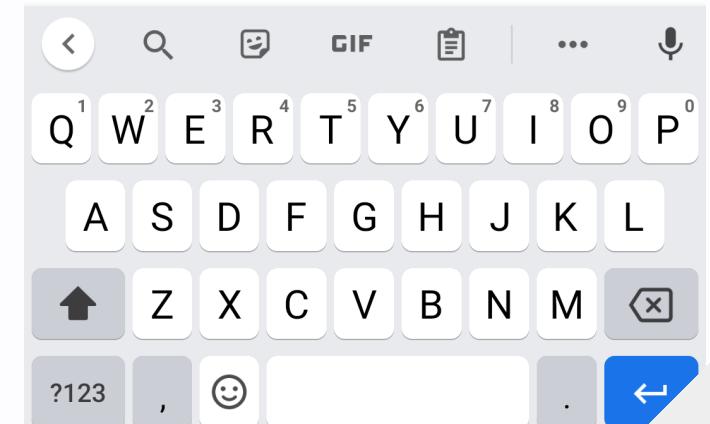
quod hic sed harum vel eveniet totam  
facilis maiores ducimus optio quae ut  
quis corrupti et neque nihil neque ea nihil  
illum illo voluptatem iure consectetur au



ex odio et consequatur corporis inventore  
sit alias molestiae in minus asperiores  
qui rerum laboriosam qui ea vel sunt  
unde voluptatum ut ratione placeat aut hi



Hello World!



# Zostawmy na chwilę apkę mobilną

# Backend

Oficjalnie mamy wyboru mamy trzy pluginy:

- @nrwl/express
- @nrwl/nest
- @nrwl/node

# Backend

Napisalem w `express.js` za pomoca metodyki TDD (tutaj chcialem byc pewny, ze wszystko dziala jak nalezy)

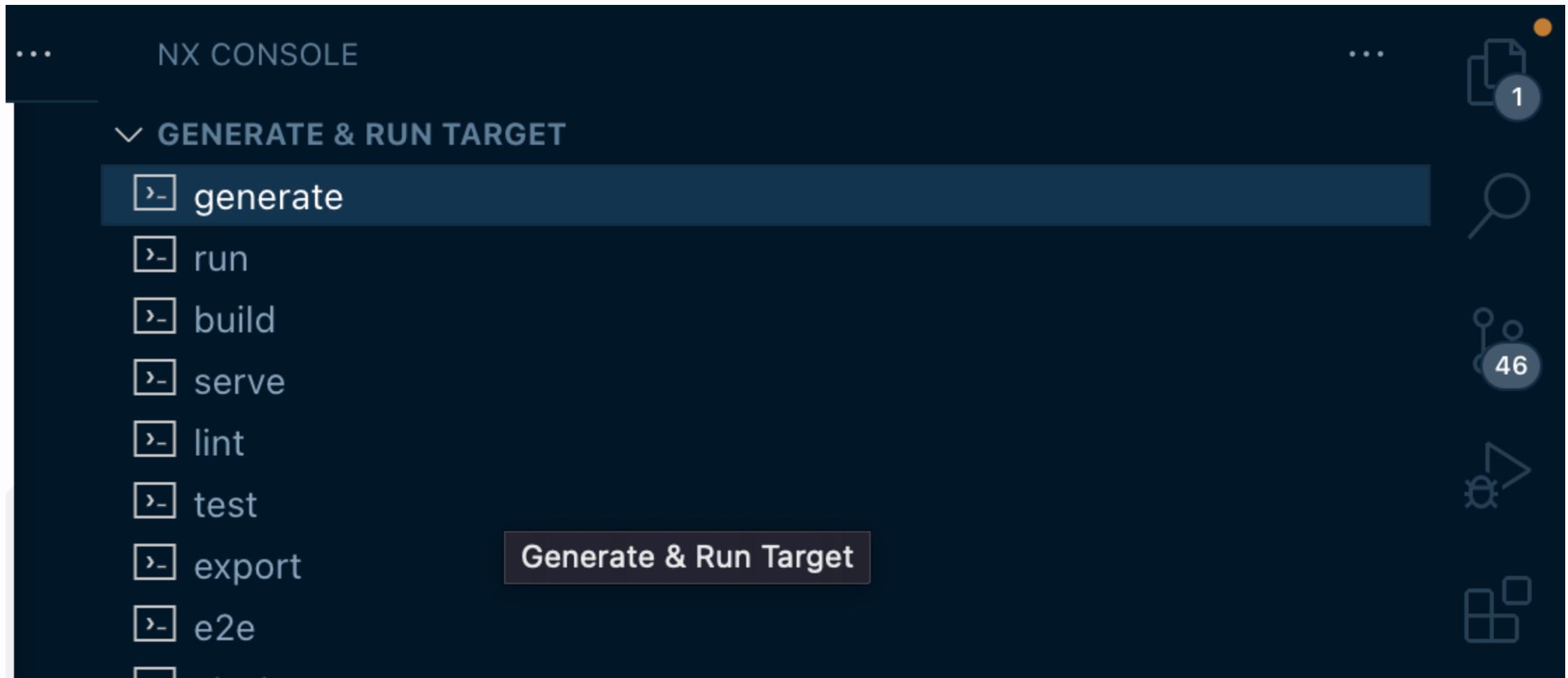
# Instaluję kolejny generator:

```
npm install --save-dev @nrwl/express
```

i generuję apke za pomocą CLI 

```
nx g @nrwl/express:app api
```

albo wyklikuję z UI 😎



Run**Name**

Directory

Babel Jest

Frontend Project

Js

Linter

Pascal Case Files

Set Parser Options Project

Skip Format

Skip Package Json

Standalone Config

Tags

Unit Test Runner

**nx generate @nrwl/express:application** 

api

**name**

The name of the application.

api

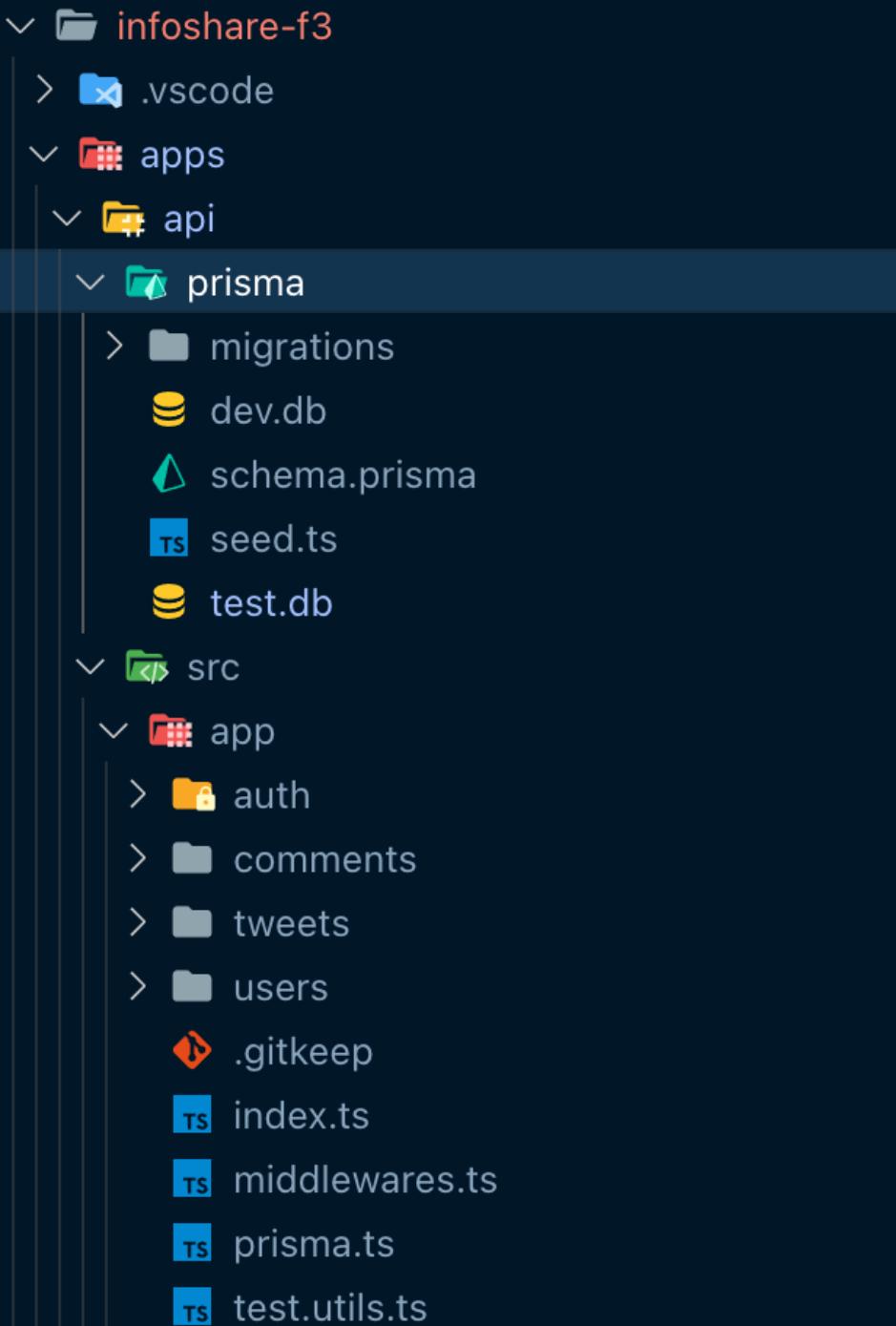
**directory**

The directory of the new application.

**babelJest** Use `babel` instead `ts-jest`.**frontendProject**

Frontend project that needs to access this application. This sets up proxy configuration.

**js** Generate JavaScript files rather than TypeScript files.



Ta aplikacja, z perspektywy Nx nie ma wyodrębnionych bibliotek, ale zdecydowanie jest podzielona na moduły, które można by wydzielić do osobnych bibliotek.

```
import { Request, Response } from 'express';
import { Tweet } from '@infoshare-f3/types'
import { prisma } from '../prisma';

export const tweetsController = {
  getAll: async (req: Request, res: Response<{ tweets: Tweet[] }>) => {
    const { page = '1' } = req.query;
    const parsedPage = Number(page);
    const tweets = await prisma.tweet.findMany({
      //...
    });
    return res.status(200).json({
      tweets,
    });
  },
  // ...
}
```

## Przykład kontrolera

# Klient API

Po napisaniu backendu napisalem bibliotekę, która jest klientem do tego api

```
import axios from 'axios';
import { User, Comment, Tweet } from '@infoshare-f3/types';

export const instance = axios.create({
  baseURL: 'http://localhost:3333/',
});

export const client = {
  auth: {
    login: async ({...}),
    logout: () => {...},
  },
  tweets: {
    getAll: async ({ page = 1 }): { page: number } => {...},
    create: async ({ text }): { text: string } => {...},
    delete: async ({ tweetId }): { tweetId: string } => {...},
    like: async ({ tweetId }): { tweetId: string } => {...},
    unlike: async ({ tweetId }): { tweetId: string } => {...},
  },
  comments: {
    getAll: async ({ tweetId }): { tweetId: string } => {...},
    create: async ({ tweetId, text }): { tweetId: string; text: string } => {...},
  },
  users: {
    getAll: async () => {...},
    delete: async (userId: string) => {...},
  },
};
```

Biblioteka jest zbiorem gotowych funkcji, dzięki którym komunikujemy się z API. Jeśli dorzucimy do niej typy to mamy perfekcyjną warstwę komunikacji serwer klient.

# Data provider

`react-query` do fetchowania i cache'owania danych.  
Można reużywać między aplikacjami.

```
const loginMutation = useMutation(client.auth.login, {
  onSuccess: (response) => {
    if (response.status === 200) {
      queryClient.setDefaultOptions({
        queries: {
          enabled: true,
        },
      });
      queryClient.refetchQueries();
      const user = jwtDecode(response.data.token) as Me;
      setMe(user);
    }
  },
});
```

```
const tweetsQuery = useInfiniteQuery(
  'tweets',
  ({ pageParam = 1 }) => {
    return client.tweets.getAll({ page: pageParam });
  },
  {
    getNextPageParam: (lastPage: any, allPages: any) => {
      const hasNextPage =
        lastPage?.data?.tweets?.length % 10 === 0 &&
        lastPage?.data?.tweets?.length !== 0;
      const pages = allPages.length;
      if (hasNextPage) {
        return pages + 1;
      }
    },
  },
);
```

```
export const useCommentsForTweetQuery = (tweetId?: string) => {
  return useQuery(['commentsForTweet', tweetId], () => {
    if (tweetId) {
      return client.comments.getAll({ tweetId });
    }

    return null
  }, {
    enabled: !!tweetId
  });
};
```

```
export const useDeleteUserMutation = () => {
  const queryClient = useQueryClient()
  return useMutation(client.users.delete, {
    onSuccess: (response) => {
      if (response.status === 200) {
        queryClient.invalidateQueries('users')
      }
    }
  })
}
```

# Kończenie aplikacji mobilnej

Wszystkie ekranы aplikacji mobilnej wyciągnąłem do zewnętrznego modułu `native-screens`. Dzięki NX wygenerowanie nowego modułu jest banalnie proste.

```
nx g @nrwl/react-native:library native-screens
```

# Aplikacja webowa

wygenerowałem aplikacje webową za pomocą generatora Nx React. Użyłem gotowych komponentów z aplikacji mobilnej.

● ○ ● TweeterWeb X +

localhost:4200 Guest :

Co się dzieje?

Tweet

@Cathrine27 · today at 11:13 AM  
architecto quis dolore hic et eum provident qui suscipit quia expedita cumque molestias voluptatem sed quo omnis omnis est consequuntur sed sapiente odit exerci

12 29

@Marisa\_Emmerich · today at 10:54 AM  
maxime molestiae asperiores maxime delectus ipsam in asperiores quisquam delectus vel est porro sunt molestiae tenetur omnis non illo rerum sed neque consequunt

13 30

@Stanford33 · today at 10:16 AM  
asperiores pariatur ipsa occaecati omnis nobis minus blanditiis eos ullam quia voluptate quea veniam quis eum temporibus deserunt omnis libero atque accusamus a

14 30

@Constantin35 · today at 10:12 AM  
soluta sed sunt placeat et magnam explicabo repellendus recusandae sed perferendis exercitationem laborum quaerat cumque quia quis aliquam assumenda animi bland

● ○ ● TweeterWeb X +

localhost:4200/tweets/cl38kchw8002k5ncvgmnl0vfr Guest :

@Cathrine27 · today at 11:13 AM  
architecto quis dolore hic et eum provident qui suscipit quia expedita cumque molestias voluptatem sed quo omnis omnis est consequuntur sed sapiente odit exerci

12 29

Skomentuj tweeta

Skomentuj

corrupti fuga praesentium aperiam officia iste repudiandae perspiciatis a voluptatum cum fuga rerum dicta non quos omnis sequi et sunt vero non nisi tempore aut

ducimus natus maiores harum ea nemo suscipit et sit adipisci totam quae fuga recusandae veniam minus temporibus excepturi pariatur voluptas quod culpa et non se

ducimus quis et consequuntur ratione in perspiciatis nulla voluptas at sit minus ea fuga dolorem accusamus dolorem autem veniam voluptate corrupti quis velit si

fuga numquam eos iste voluptatem ab officia iure vero ratione consequatur dolorum a optio ut enim eius commodi recusandae iure aut nostrum est et natus consequu

est at facilis consequatur quia qui dolores voluptatum velit natus sed at dolores

# Panel administradora

Welcome to tweeter-admin! X

localhost:4300/users Guest ⋮

 Hilda Upton (@Ward34) <a href="#">Delete</a>	 Lawrence Marquardt (@Celestino.Swift79) <a href="#">Delete</a>	 Gwen McGlynn (@Michele81) <a href="#">Delete</a>
 Julio Welch (@Buster.Schroeder) <a href="#">Delete</a>	 Leroy Buckridge (@Maxime_Runolfsdottir94) <a href="#">Delete</a>	 Jacqueline Stokes (@Nico.Marvin) <a href="#">Delete</a>
 Darryl Harber (@Janice.Sauer) <a href="#">Delete</a>	 Derrick Douglas (@Rose.Yost66) <a href="#">Delete</a>	 Ms. Kirk Mills (@Delaney_Lind61) <a href="#">Delete</a>
 Kellie O'Hara I (@Wendell.Waters) <a href="#">Delete</a>	 Helen Gerhold (@Amiya.Gusikowski) <a href="#">Delete</a>	 Wade Labadie (@Jarrod.Hahn) <a href="#">Delete</a>
 Spencer Champlin (@Terrell_Klocko45) <a href="#">Delete</a>	 Manuel Beatty (@Stanford33) <a href="#">Delete</a>	 Dr. Lucy Rau (@Otto.Oberbrunner17) <a href="#">Delete</a>

Welcome to tweeter-admin! X +

localhost:4300/tweets Guest :

@Cathrine27 · today at 11:13 AM  
architecto quis dolore hic et eum provident qui suscipit  
quia expedita cumque molestias voluptatem sed quo  
omnis omnis est consequuntur sed sapiente odit exerci

12 29

@Marisa\_Emmerich · today at 10:54 AM  
maxime molestiae asperiores maxime delectus ipsam in  
asperiores quisquam delectus vel est porro sunt  
molestiae tenetur omnis non illo rerum sed neque  
consequunt

13 30

**Delete tweet**

@Stanford33 · today at 10:16 AM  
asperiores pariatur ipsa occaecati omnis nobis minus  
blanditiis eos ullam quia voluptate quae veniam quis eum  
temporibus deserunt omnis libero atque accusamus a

14 30

@Constantin35 · today at 10:12 AM  
soluta sed sunt placeat et magnam explicabo repellendus  
recusandae sed perferendis exercitationem laborum  
quaerat cumque quia quis aliquam assumenda animi  
bland

10 33

**Delete tweet**

@Madisen\_Glover · today at 9:52 AM  
est sint eveniet mollitia provident nesciunt odio  
dignissimos consequatur et et corrupti magni odit nisi  
velit velit doloribus quod aliquam omnis delectus expli

10 33

@Jarrod.Hahn · today at 9:26 AM  
voluptate quaerat autem similiqe dolor aut maxime ut  
accusamus quos distinctio nostrum repudiandae  
voluptatem qui incident ea perferendis eos expedita  
perspici

14 26

**Delete tweet**

@Brandon90 · today at 8:54 AM  
dolor aliquam necessitatibus dolorem autem atque dolor et

@Constantin35 · today at 8:21 AM  
voluntas voluntas molestias itaque quo est et sint illum

# **Co mozna bylo jeszcze wspoldzielić poza częściami UI?**

- miedzy wszystkimi aplikacjami - typy, helpery, schemas
- miedzy wszystkimi aplikacjami UI - propsy komponentów, logika formularzy, data-providers

```
export const createTweetSchema = object({
  text: string()
    .required('text is a required field')
    .max(160, 'text must be at most 160 characters'),
}).required();

export const createTweetSchemaExpress = object({
  body: createTweetSchema,
}).required();
```

```
import { createTweetSchemaExpress } from '@infoshare-f3/schemas';

export const router = Router();

router.get('/tweets', authenticateToken, tweetsController.getAll);
router.post(
  '/tweets',
  authenticateToken,
  validate(createTweetSchemaExpress),
  tweetsController.create
);
```

```
import { SubmitHandler, useForm } from 'react-hook-form';
import { yupResolver } from '@hookform/resolvers/yup';
import { createTweetSchema, commentTweetSchema } from '@infoshare-f3/schemas';
import { CreateComment } from '@infoshare-f3/types';
import { useTweetsContext, useCommentTweetMutation } from '@infoshare-f3/data-providers';

export const useCreateTweetForm = () => {
  const { createTweetMutation } = useTweetsContext();
  const createTweetForm = useForm<CreateComment>({
    resolver: yupResolver(createTweetSchema),
    defaultValues: {
      text: '',
    },
    mode: 'all',
  });

  const handleValid: SubmitHandler<CreateComment> = (data) => {
    createTweetMutation?.mutate(data);
  };

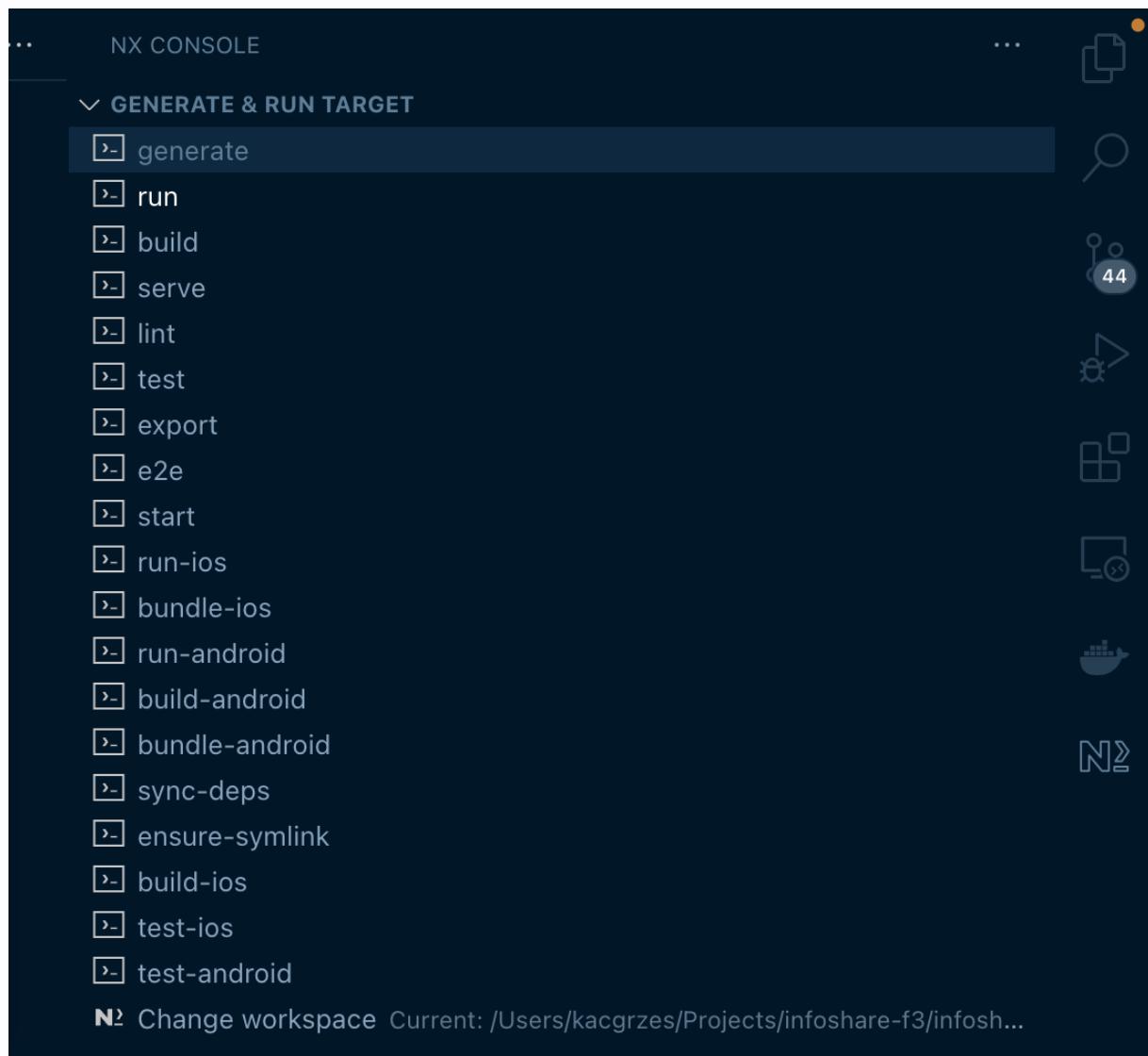
  const onSubmit = createTweetForm.handleSubmit(handleValid);

  return {
    ...createTweetForm,
    onSubmit,
  };
};
```

# Reguła której warto zapamiętać

- 80% logiki idzie do katalogu `libs/`
- 20% do katalogu `apps/`
- `libs/` to moduły naszych aplikacji ustrukturyzowane w
- `apps/` to kontenery na te moduły

# **Polecenia dostępne w NX z poziomu VSCode**



# **Uruchamianie wielu projektów jednocześnie**

└ test-ios

└ COMMON NX COMMANDS

└ dep-graph

└ run-many

└ affected

└ run-many

└ affected:apps

└ affected:build

└ affected:dep-graph

└ affected:e2e

└ affected:libs

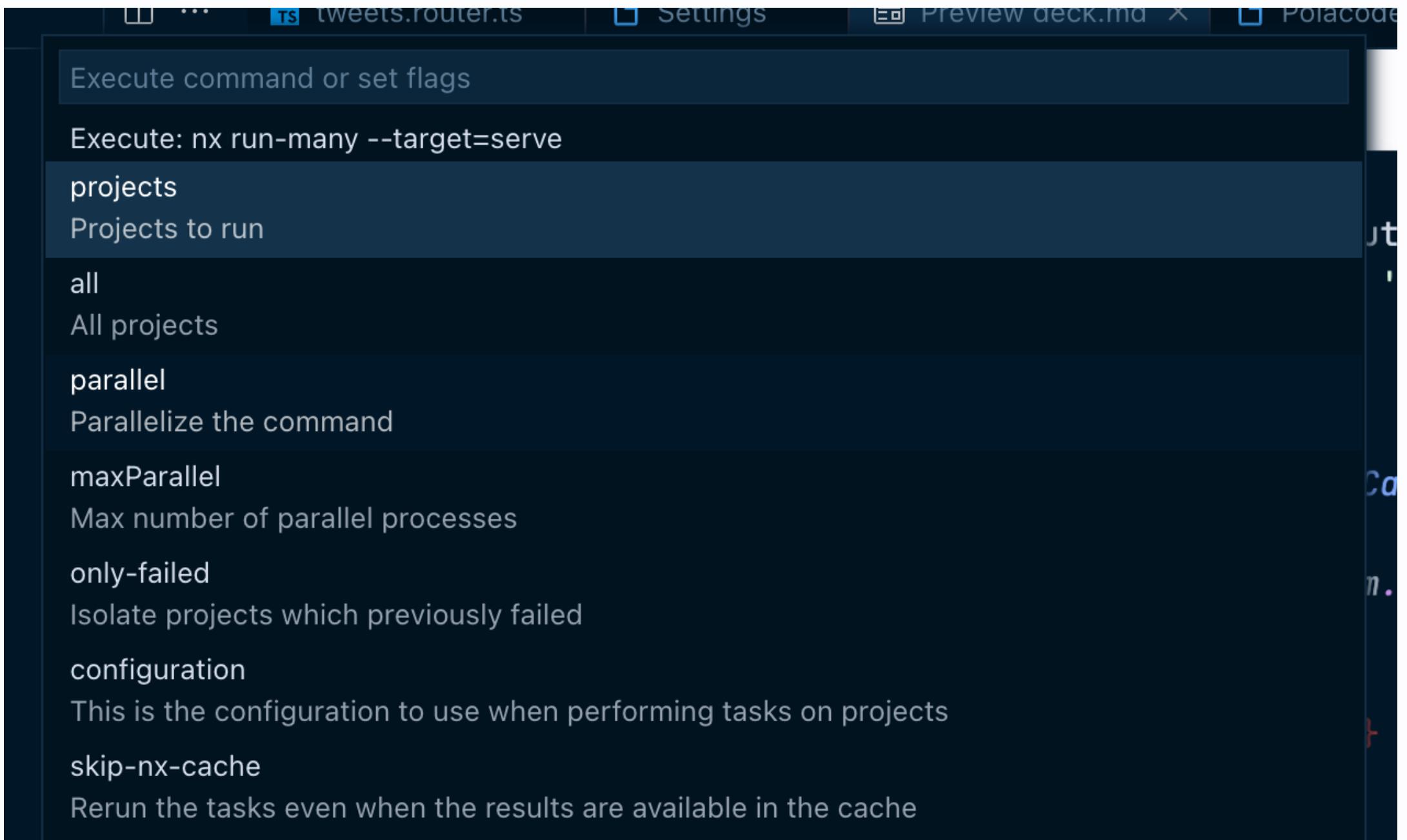
└ affected:lint

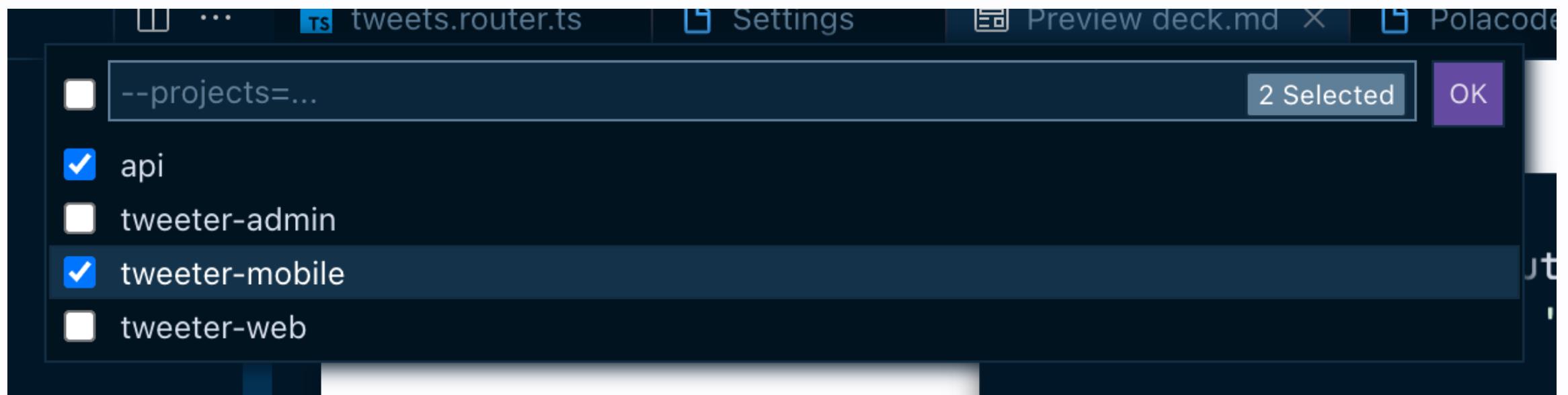
└ affected:test

└ list

└ migrate

```
|  
build  
build-android  
build-ios  
bundle-android  
bundle-ios  
e2e  
ensure-symlink  
export  
lint  
run-android  
run-ios  
serve  
start  
sync-deps  
test  
test-android  
test-ios
```





# Generowanie grafów zależności

*Info* Analyse and visualize your workspace.

*Filter* lib name, other lib name

**Include related libraries**

Show libraries that are related to your search.

*Eye* Show all projects

*Hide* Hide all projects

**Group by folder**

Visually arrange libraries by folders with different colors.

**Activate proximity**

Explore connected libraries step by step.

— 1 +

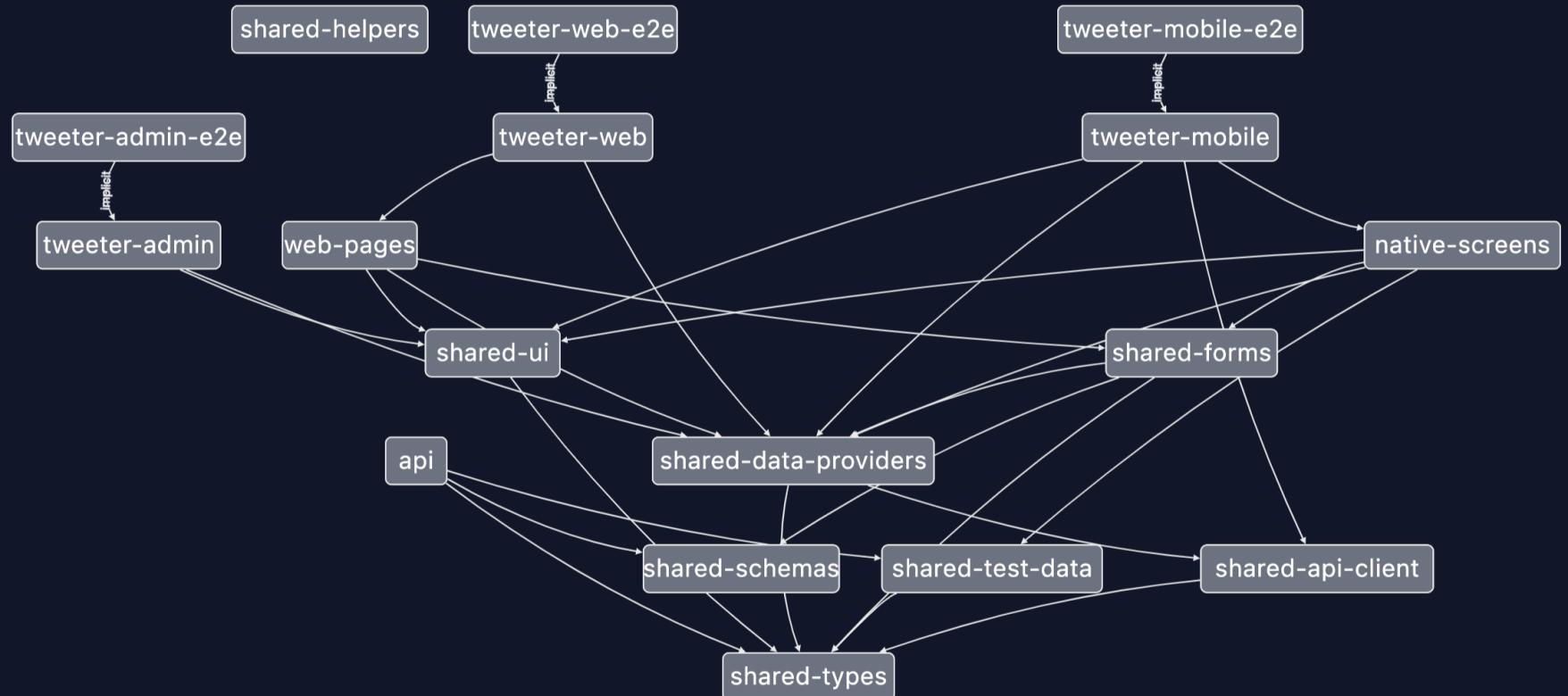
app projects



api



tweeter-admin



**i** Analyse and visualize your workspace.

lib name, other lib name

**Include related libraries**

Show libraries that are related to your search.

**Show all projects**

**Hide all projects**

**Group by folder**

Visually arrange libraries by folders with different colors.

**Activate proximity**

Explore connected libraries step by step.

-  1  +

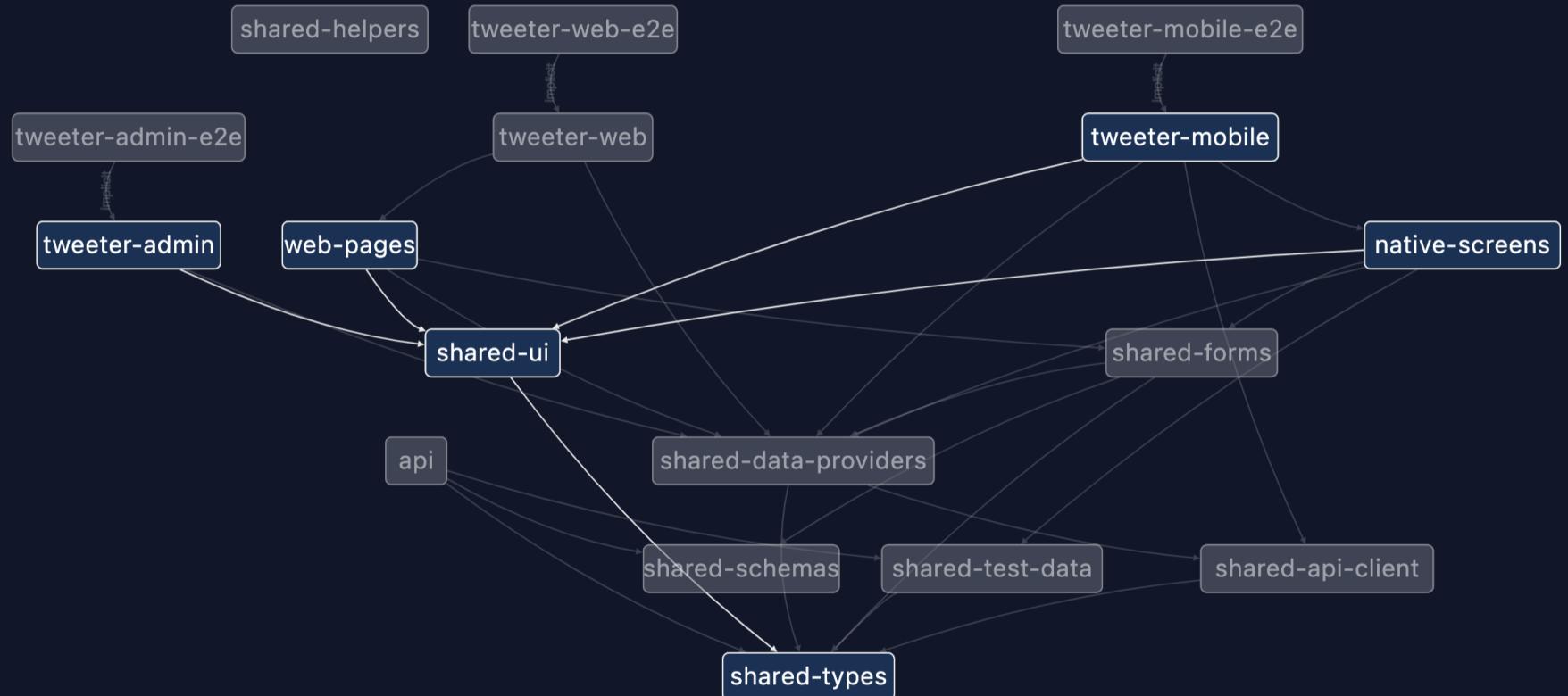
app projects



api



tweeter-admin



**i** Analyse and visualize your workspace.

lib name, other lib name

**Include related libraries**

Show libraries that are related to your search.

**Show all projects**

**Hide all projects**

**Group by folder**

Visually arrange libraries by folders with different colors.

**Activate proximity**

Explore connected libraries step by step.

-  1  +

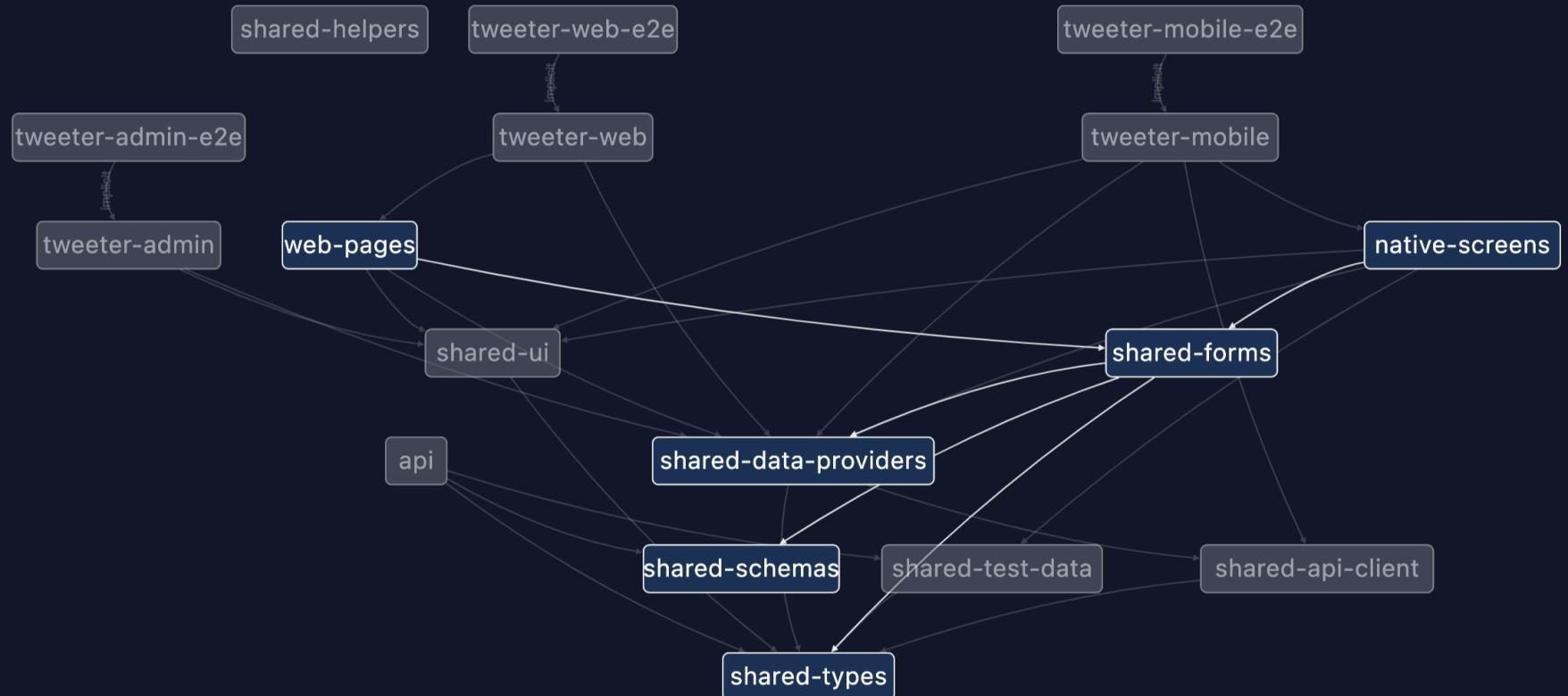
app projects



api



tweeter-admin



- Analyse and visualize your workspace.

lib name, other lib name

## Include related libraries

Show libraries that are related to your search.

 Show all projects

 Hide all projects

## Group by folder

Visually arrange libraries by folders with different colors.

## Activate proximity

Explore connected libraries step by step.

— 1 +

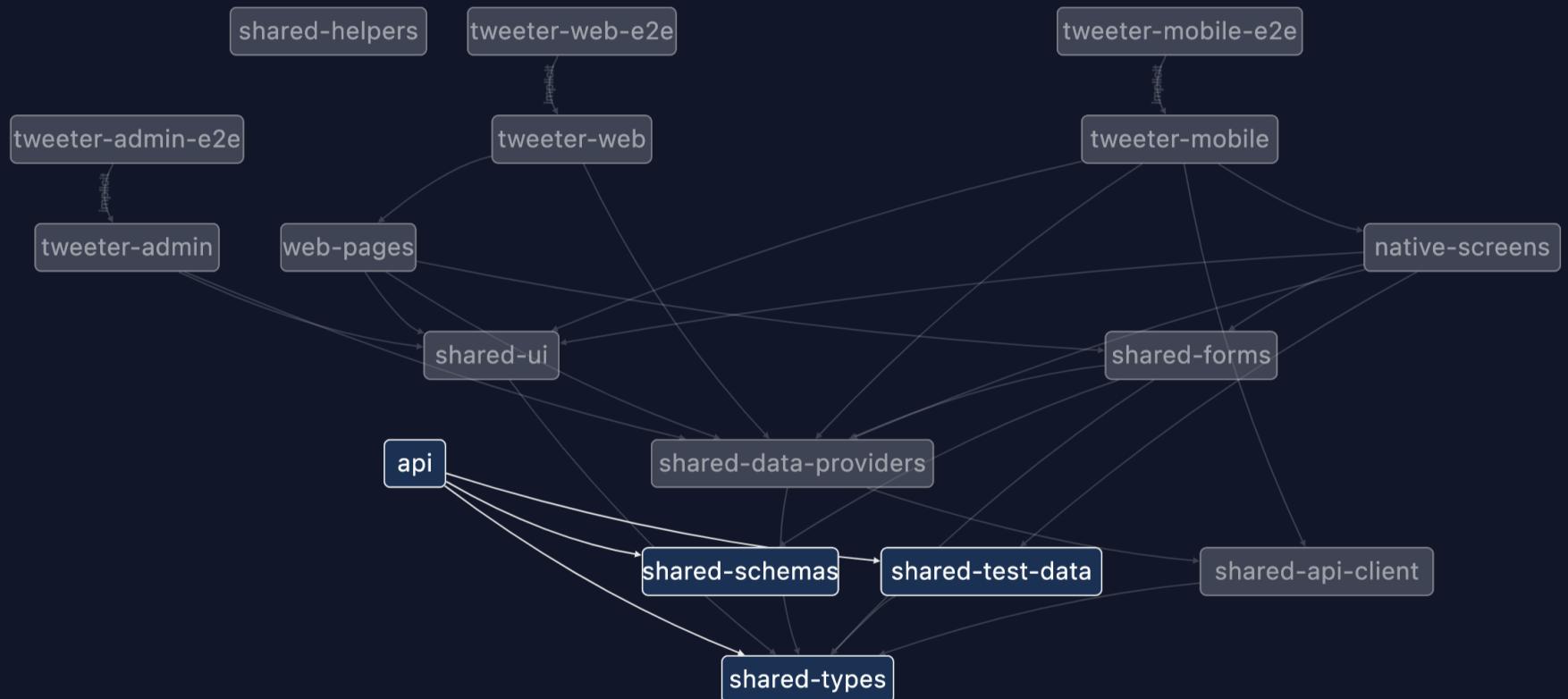
## app projects



api



tweeter-admin



**Include related libraries**

Show libraries that are related to your search.

Show all projects

Hide all projects

 **Group by folder**

Visually arrange libraries by folders with different colors.

 **Activate proximity**

Explore connected libraries step by step.

— | 1 | +

---

**app projects**

 api

 tweeter-admin

 tweeter-mobile

 tweeter-web

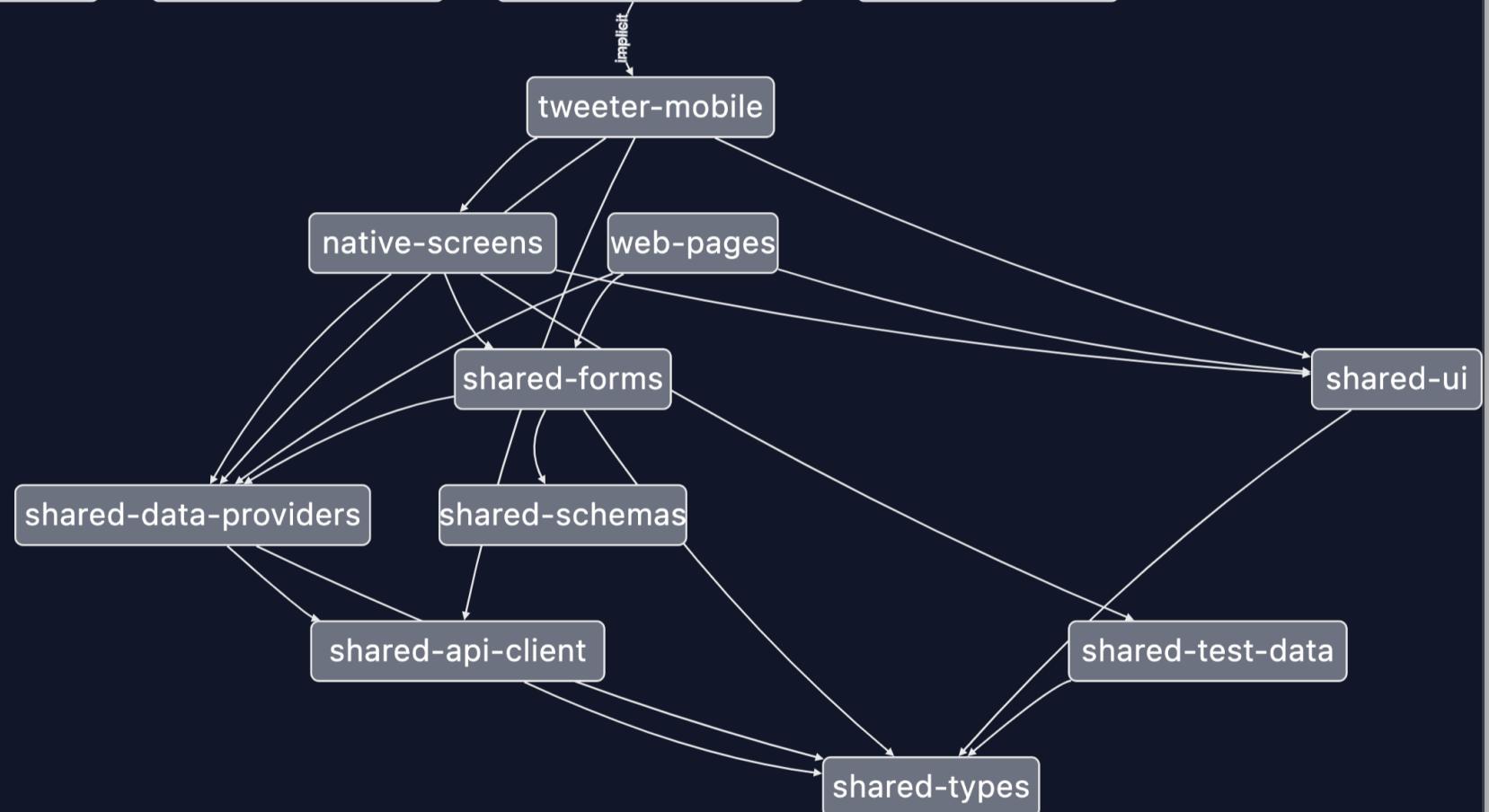


shared\helpers

tweeter-admin-e2e

tweeter-mobile-e2e

tweeter-web-e2e



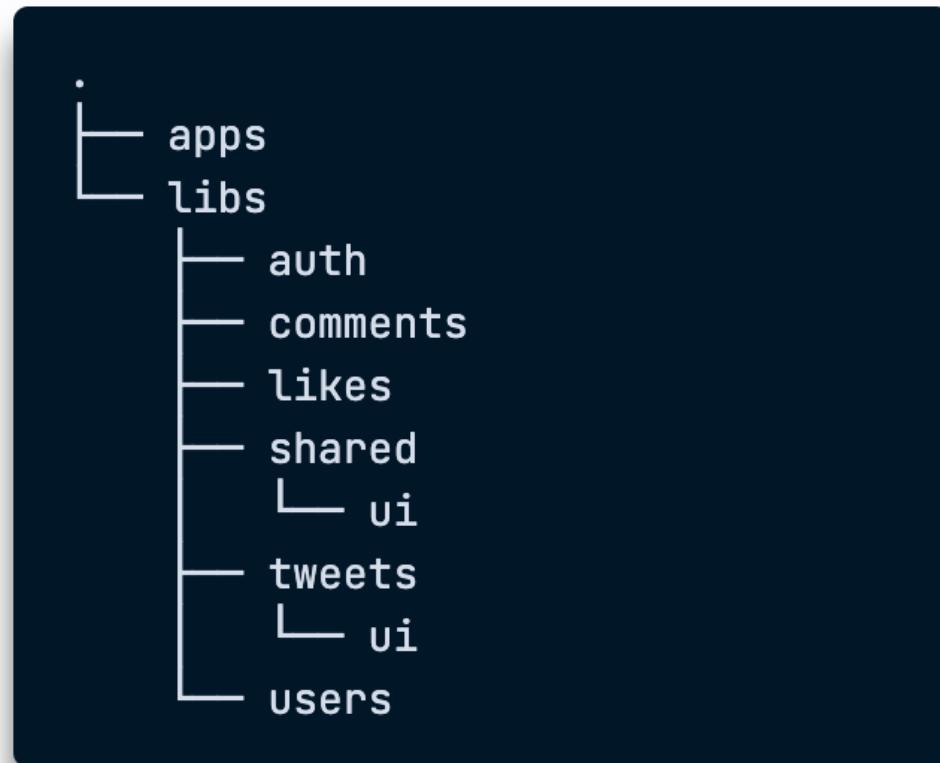
---

**e2e projects**

# co zrobiębym lepiej?

- zacząłbym od pisania backendu
- rozbicie na środowiska, bo miałem problem przy pisaniu testów i uruchamianiem aplikacji na środowisku developerskim

- rozbiłbym aplikacje api na biblioteki i dzieliłbym kod domenowo



- przebudowałem aplikacje webowa tak, zeby ekranы mogły być wspoldzielone między web i admin (web jest zbudowany za pomocą `create-react-app`, admin za pomocą `NextJS`)
- lepsze wspoldzielenie typów

# Dziękuję za uwagę!

Repo prezentacji i kodu projektu:



<https://github.com/kacgrzes/infoshare-f3>

zachęcam do forkowania i sprawdzania możliwości Nx