

Cześć



Kacper Grzeszczyk

kacgrzes.io / kacgrzes@gmail.com

Jestem programistą JS / TS :



- Mobile (React Native) - 70%
- Web (React) - 20%
- 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 dorzucać więcej funkcjonalności, ale celem prezentacji jest przede wszystkim pokazanie możliwości budowania projektów na platformie NX.

”

Jak podejsc do budowania wieloplatformowej aplikacji?

“ Teraz będzie trochę tekstu poprzeplatanego z obrazkami...
dla potomnych ”

<sciana–tekstu>

Sposób #1 - Polyrepo

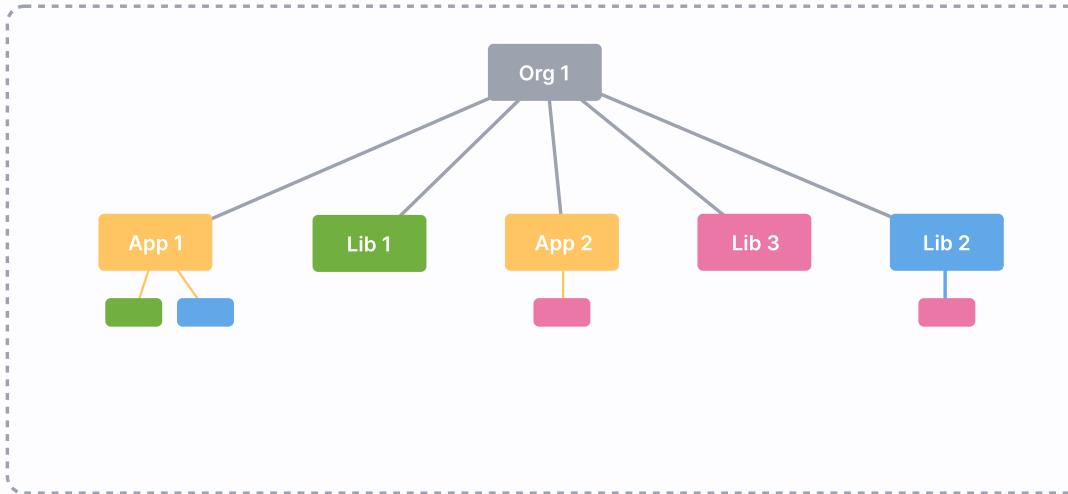
“ Kazda aplikacja, projekt ma swoje własne repozytorium. W wielu firmach w których pracowałem, to obecnie standard pisania aplikacji. Tworzy się repozytoria z monolitami. api, admin, mobile, web, każdy projekt ma swoje własne repo. Jak sa jakieś czesci wspolne to wyciaga sie je do osobnych repozytoriow.

”

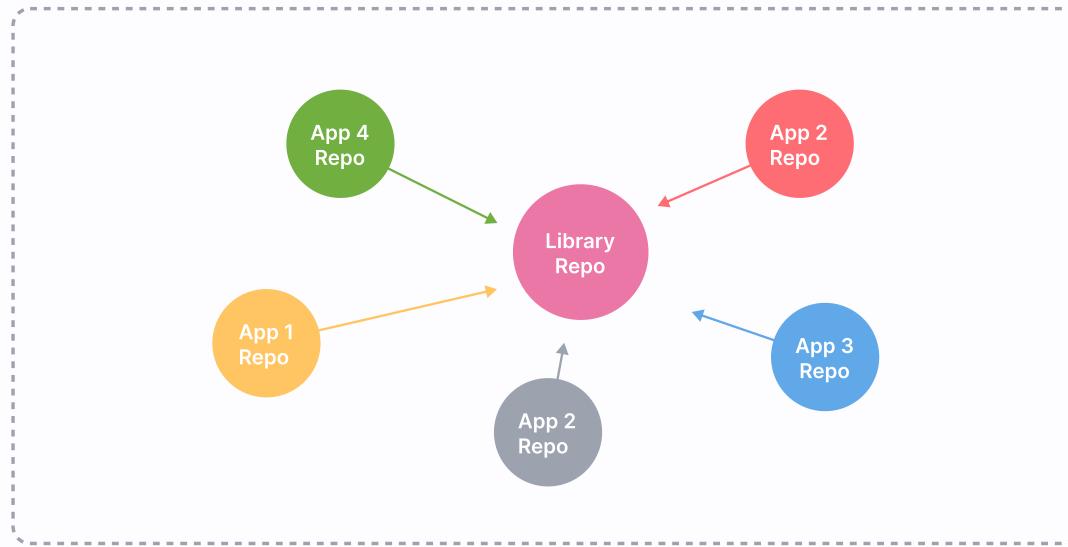
Sposób #2 – Monorepo

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

Monorepo



Polyrepo



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. ”

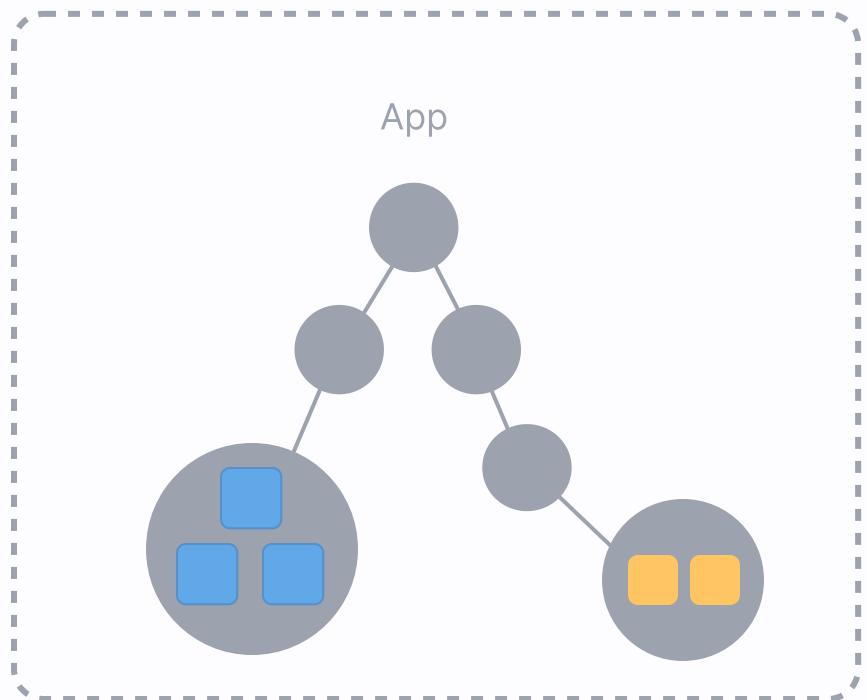
Monolit != Monorepo.

Moduł

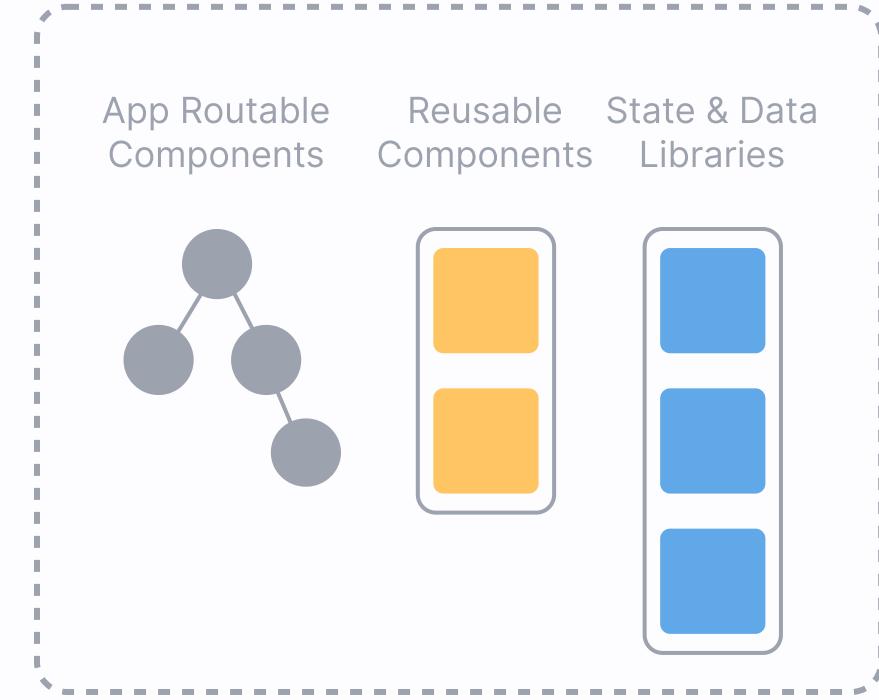
“ oddzielny twór, przeważnie w postaci osobnego pliku, zawierający zdefiniowany interfejs, a także implementacje typów wartości, klas, zmiennych, stałych oraz treści procedur i funkcji. Jest to podstawowy element koncepcji programowania modularnego pozwalający na podział kodu programu na funkcjonalne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

”

Monolith App Repo



Modular App Repo



 więcej na:
<https://monorepo.tools/>

</sciana-tekstu>

No to jedziemy, monorepo i moduły

Proces budowania aplikacji

**Określamy jakie ficzery 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

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. ”

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
```

Działanie na kodzie

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

Rozumienie kodu

```
nx graph  
nx list
```

Start 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, wiec na samym poczatku zaczalem od robienia frontu w React Native

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

Polecenie tworzy dwie aplikacje:

- tweeter-mobile
- tweeter-mobile-e2e

NX Console

Mozna uzywac wtyczki do VSCode, ktorra pozwala zapomniec o poleceniach i flagach do uzycia w konsoli

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).

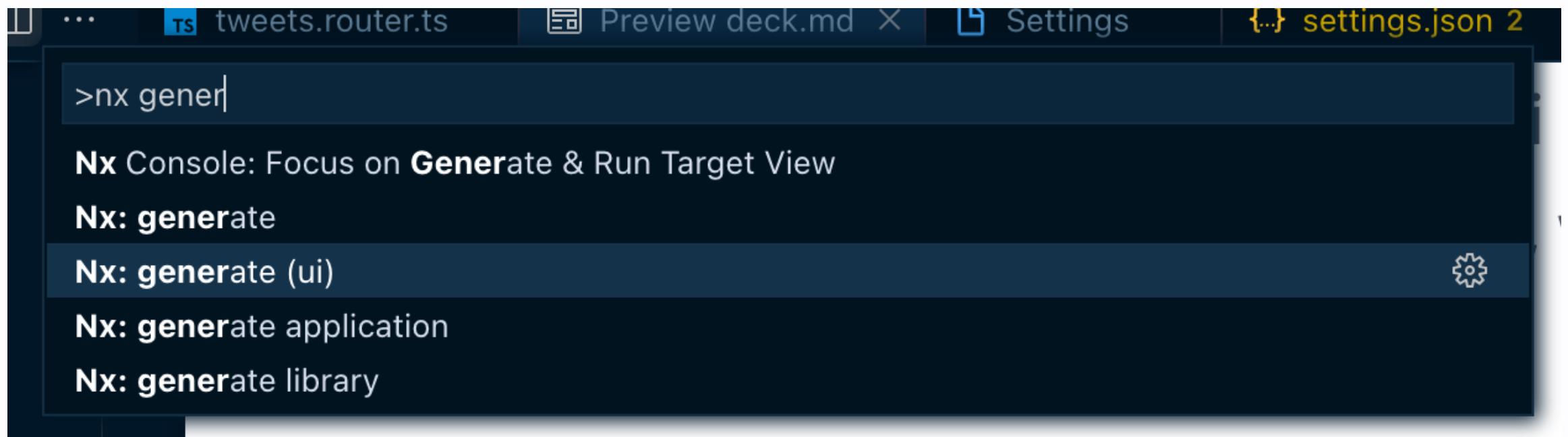
classComponent
 Use class components instead of functional component.

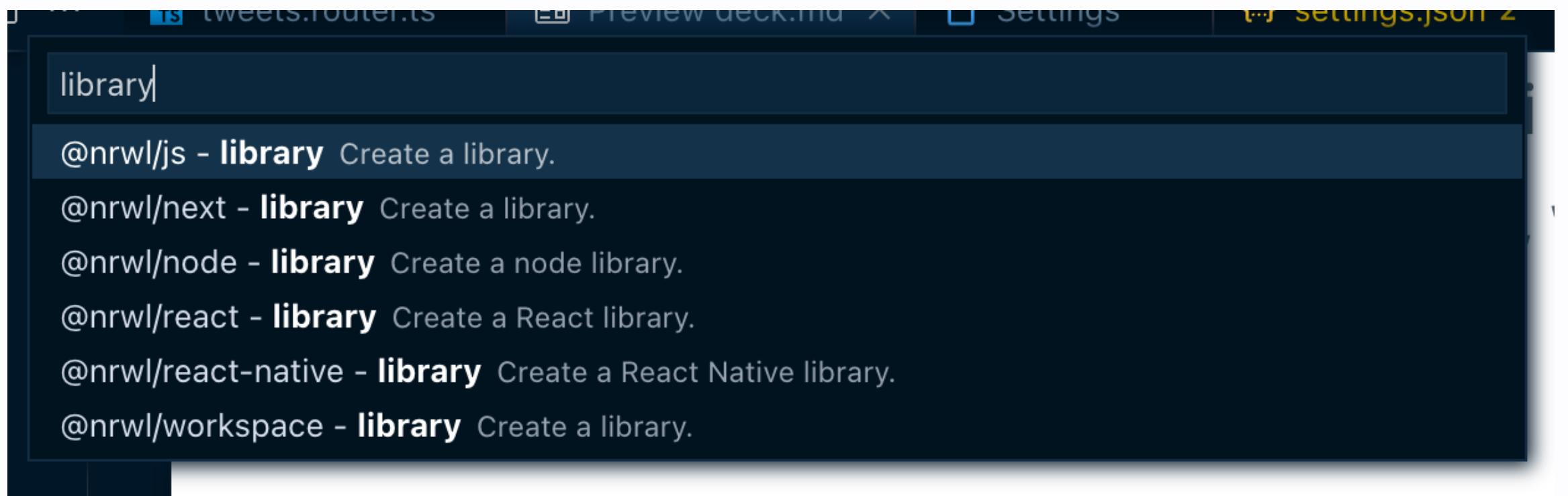
export
 When true, the component is exported from the

infoshare  binarapps 

34

W kolejnym kroku zrobilem testowe dane i skorzystałem z nich w komponentach





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

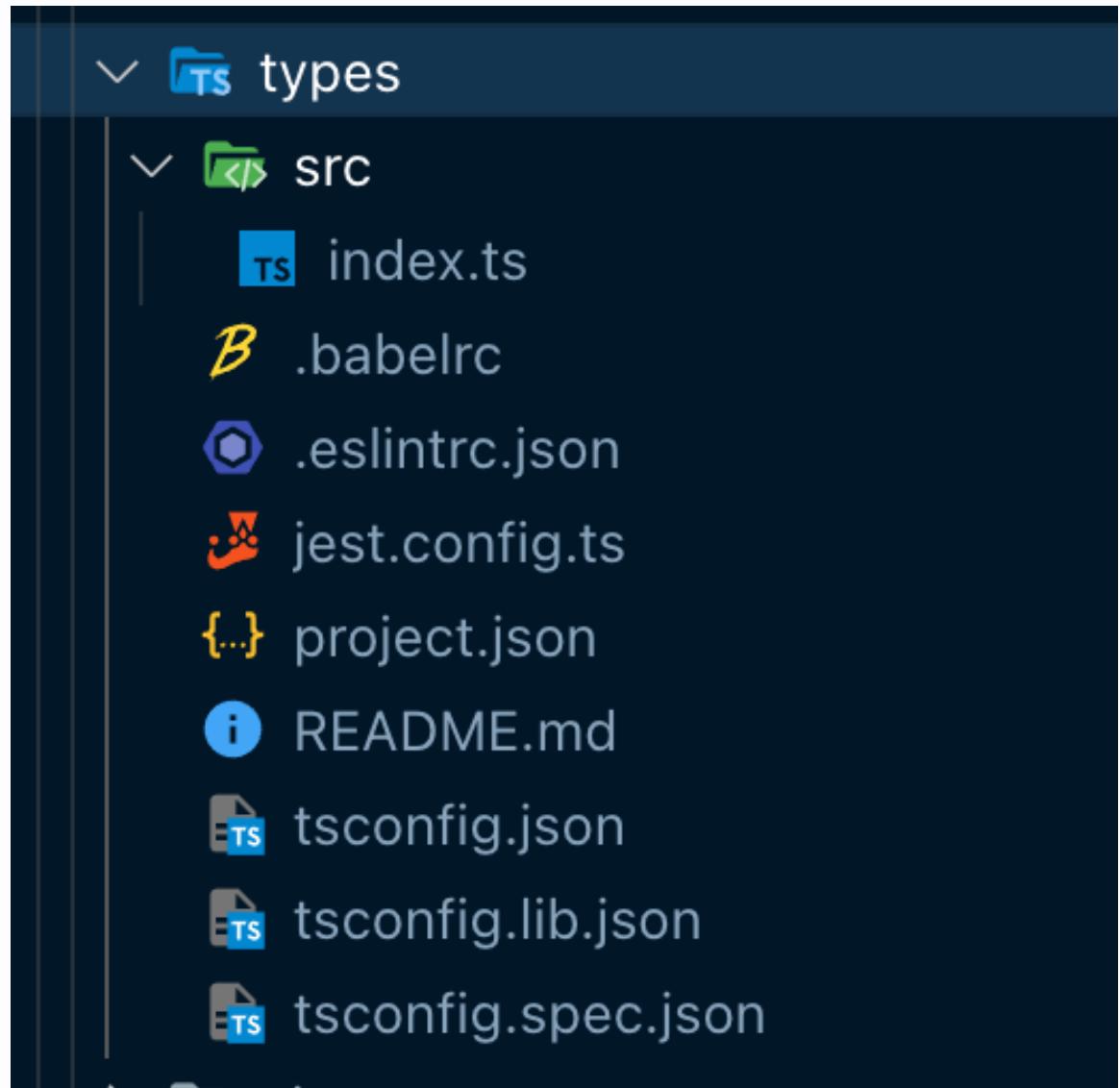


config

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

Zastanowilem się jak mozna wspoldzielic komponenty i dla samego faktu pokazania, ze mozna je wspoldzielic uznałem,
ze pojde w react-native-web + 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

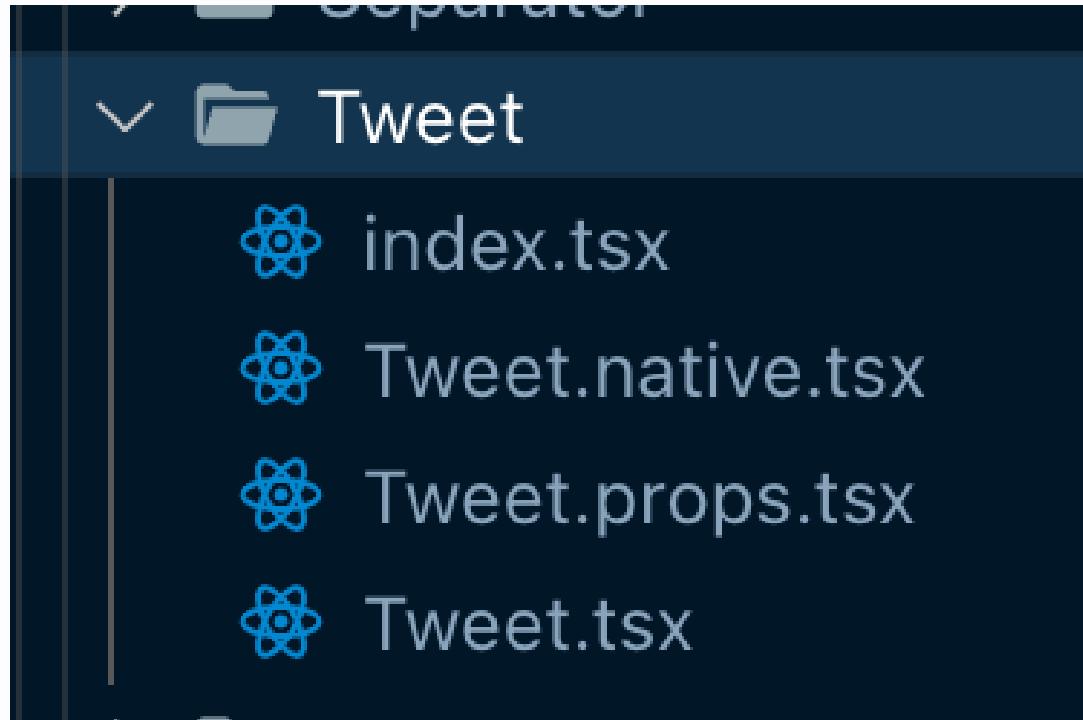
directory

```
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';
```

```
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={}
/>
```

Tworzyłem testowe dane, 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 tenet esse
explicabo si



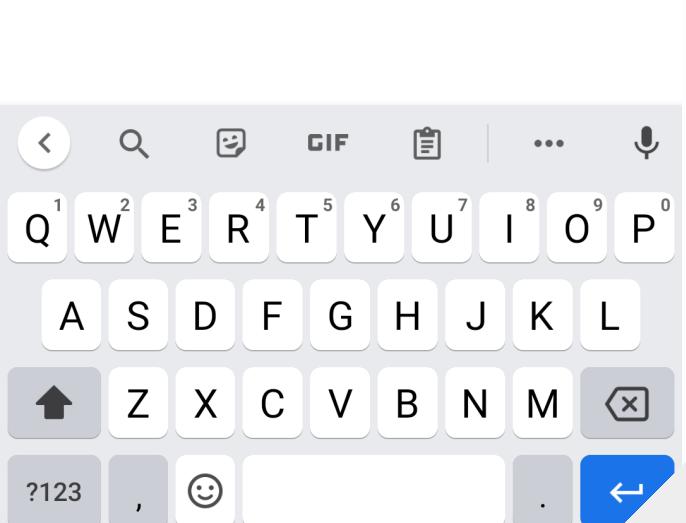
quod hic sed harum vel eveniet totam
facilis maiores ducimus optio quae ut
quis corrupti et neque nihil neque ea nihil
illum illo voluptatem iure consectetur au



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



Tweet



Zostawmy na chwilę apkę mobilną

Backend

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)

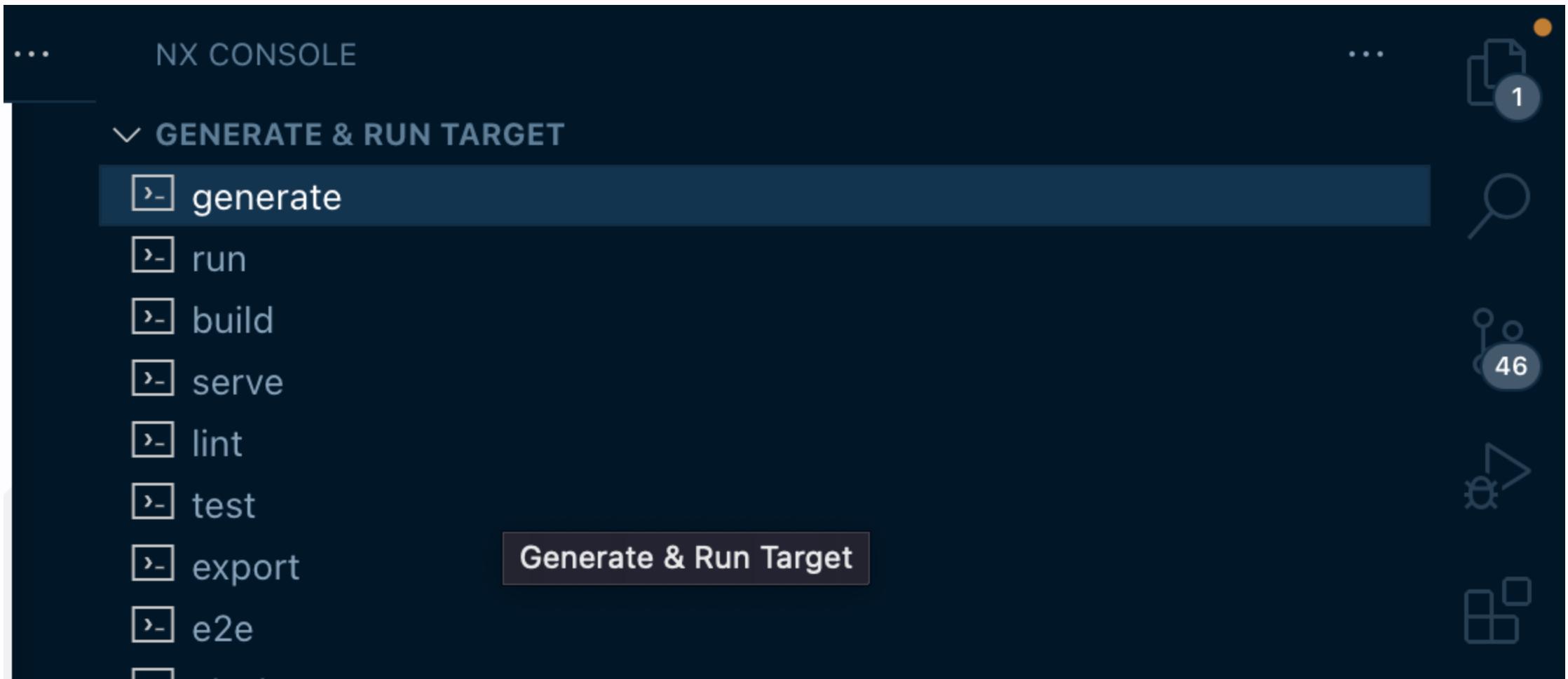
Instaluję kolejny generator:

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

i generuję aplikację za pomocą cli 

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

albo wyklikuje z UI 😎



Search flags

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

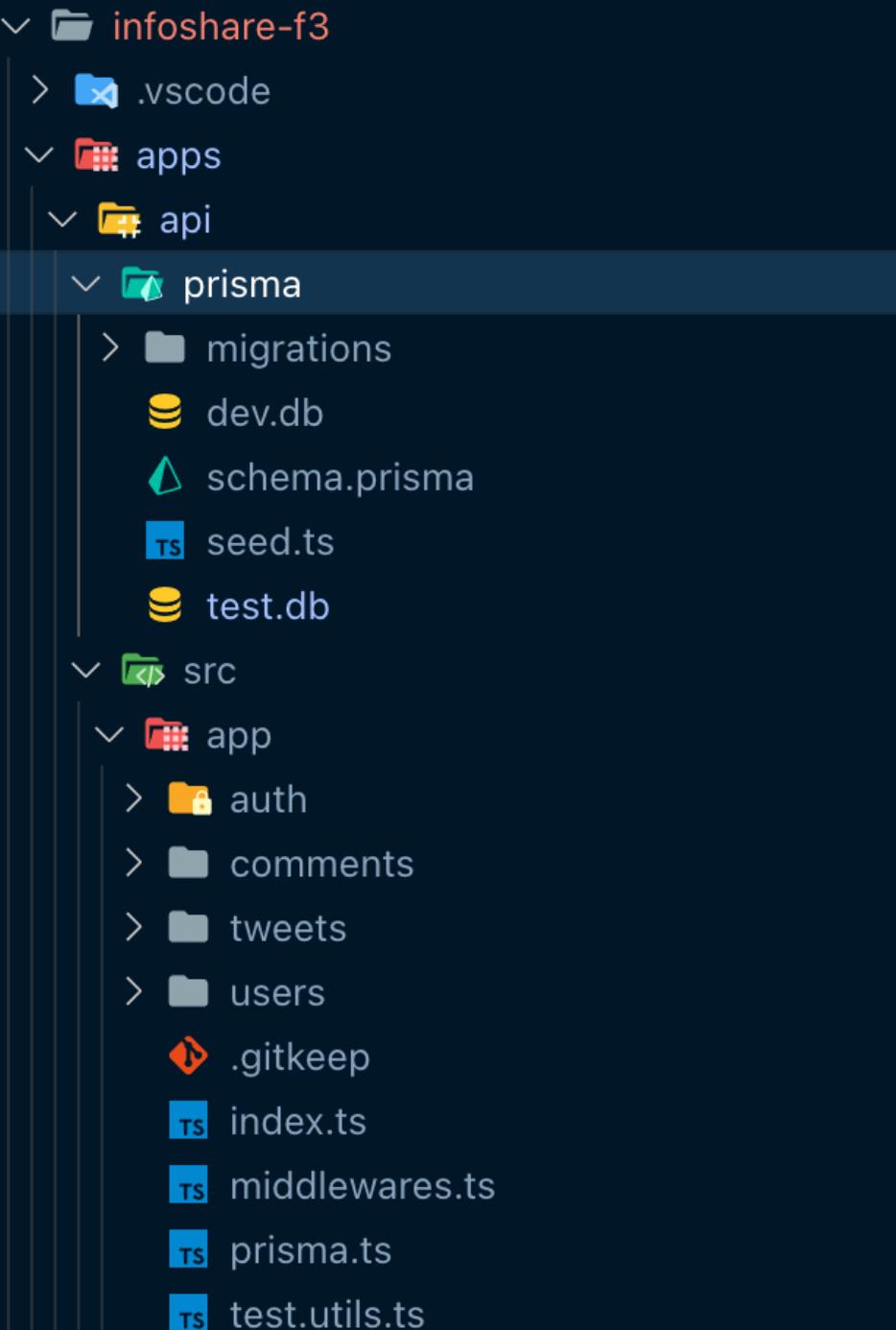
Use `babel` instead `ts-jest`.

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 nie jest monolityczna

```
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,
    });
  },
  // ...
}
```

Klient API

Po napisaniu backendu napisalem bibliotekę, ktorá je klientom 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) => {...},
  },
};
```

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
  });
};
```

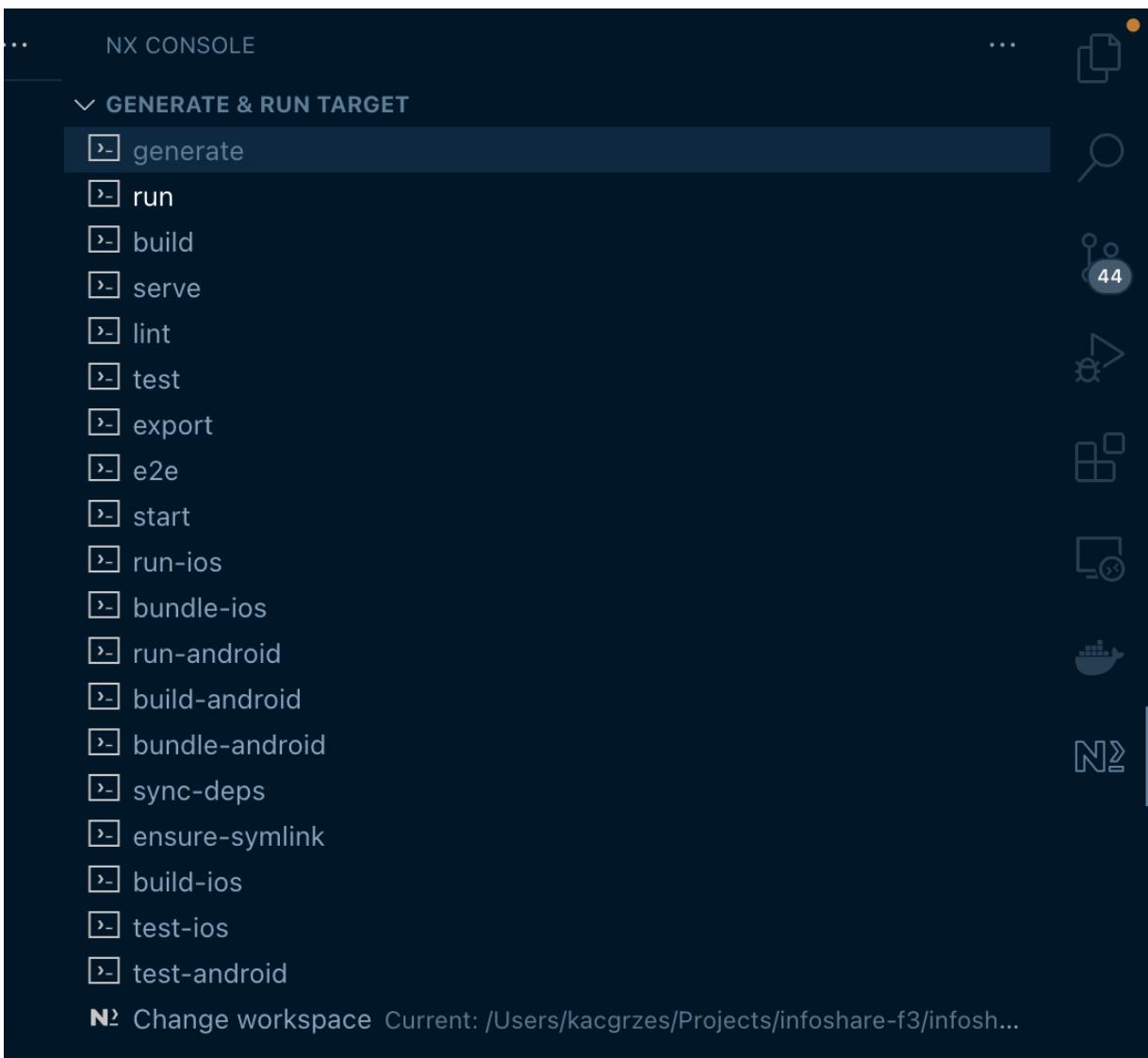
```
export const useDeleteUserMutation = () => {
  const queryClient = useQueryClient()
  return useMutation(client.users.delete, {
    onSuccess: (response) => {
      if (response.status === 200) {
        queryClient.invalidateQueries('users')
      }
    }
  })
}
```

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
```

Polecenia dostępne w NX



-
- The screenshot shows the NX CONSOLE interface with a dropdown menu open under 'COMMON NX COMMANDS' containing the following commands:
- dep-graph
 - run-many
 - affected
 - affected:apps
 - affected:build
 - affected:dep-graph
 - affected:e2e
 - affected:libs
 - affected:lint
 - affected:test
 - list
 - migrate

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

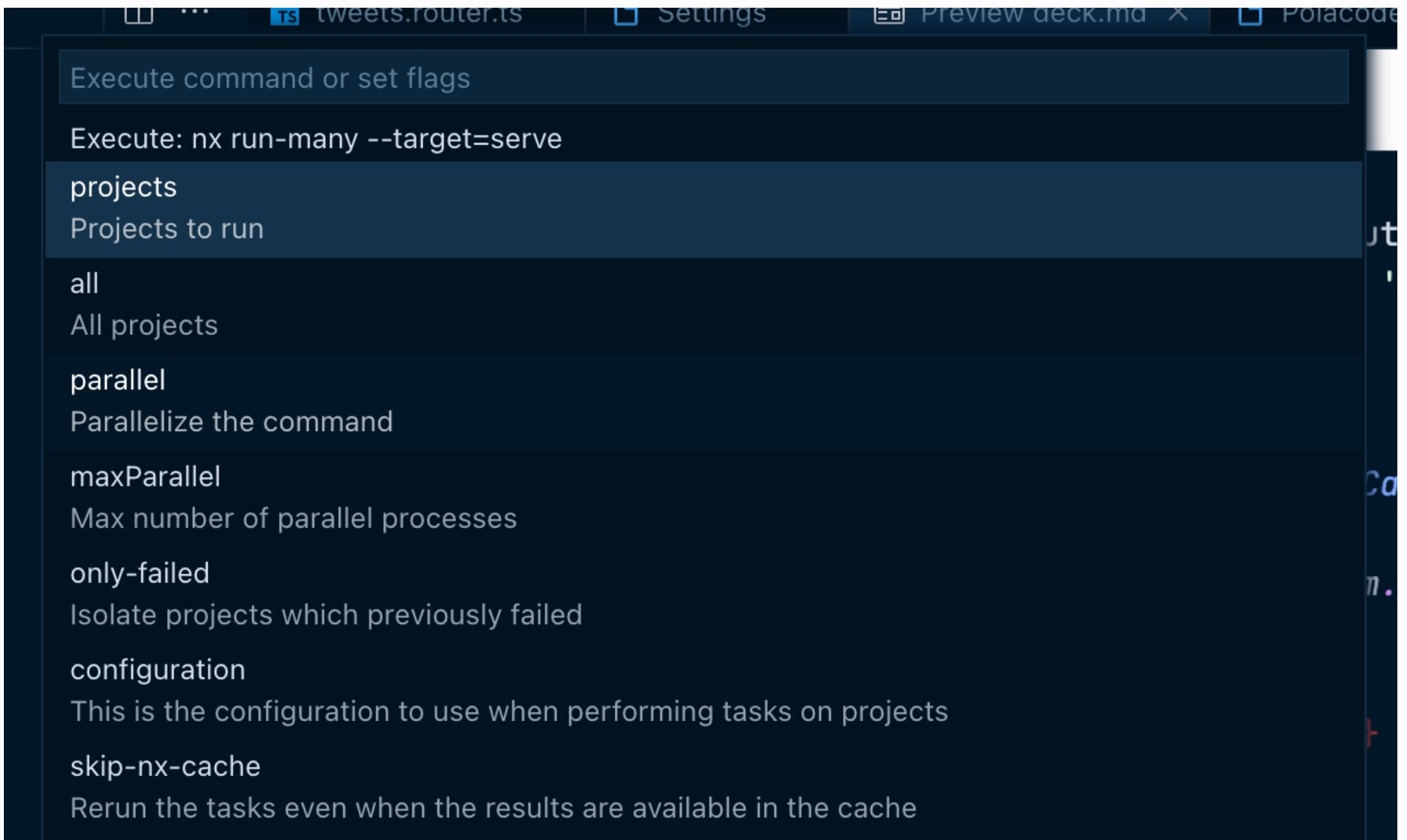
└ affected:lint

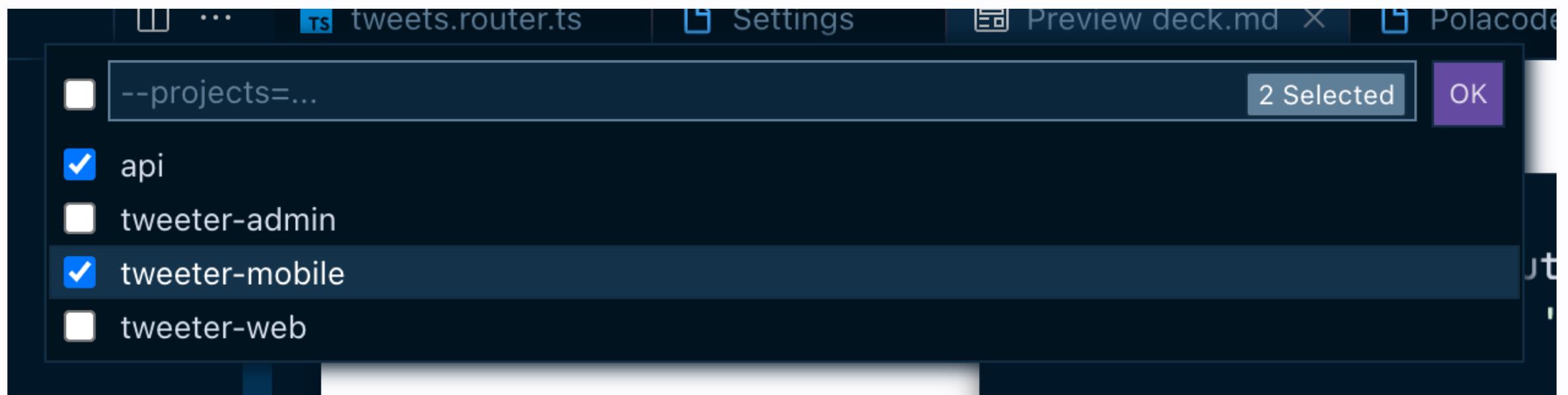
└ affected:test

└ 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
```





Kolejno

1. wygenerowałem aplikacje webowa za pomocą generatora NX React
2. oraz aplikacje admina za pomocą generatora NX NextJS

Co mozna bylo wspoldzielic?

- miedzy wszystkimi aplikacjami - typy, helpery, schemas
- miedzy wszystkimi aplikacjami UI - proste komponenty UI, propsy tych komponentów, formularze, data-providers
- api jest zbudowane w formie monolitu

Reguła której warto zapamiętać

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

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.

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 +

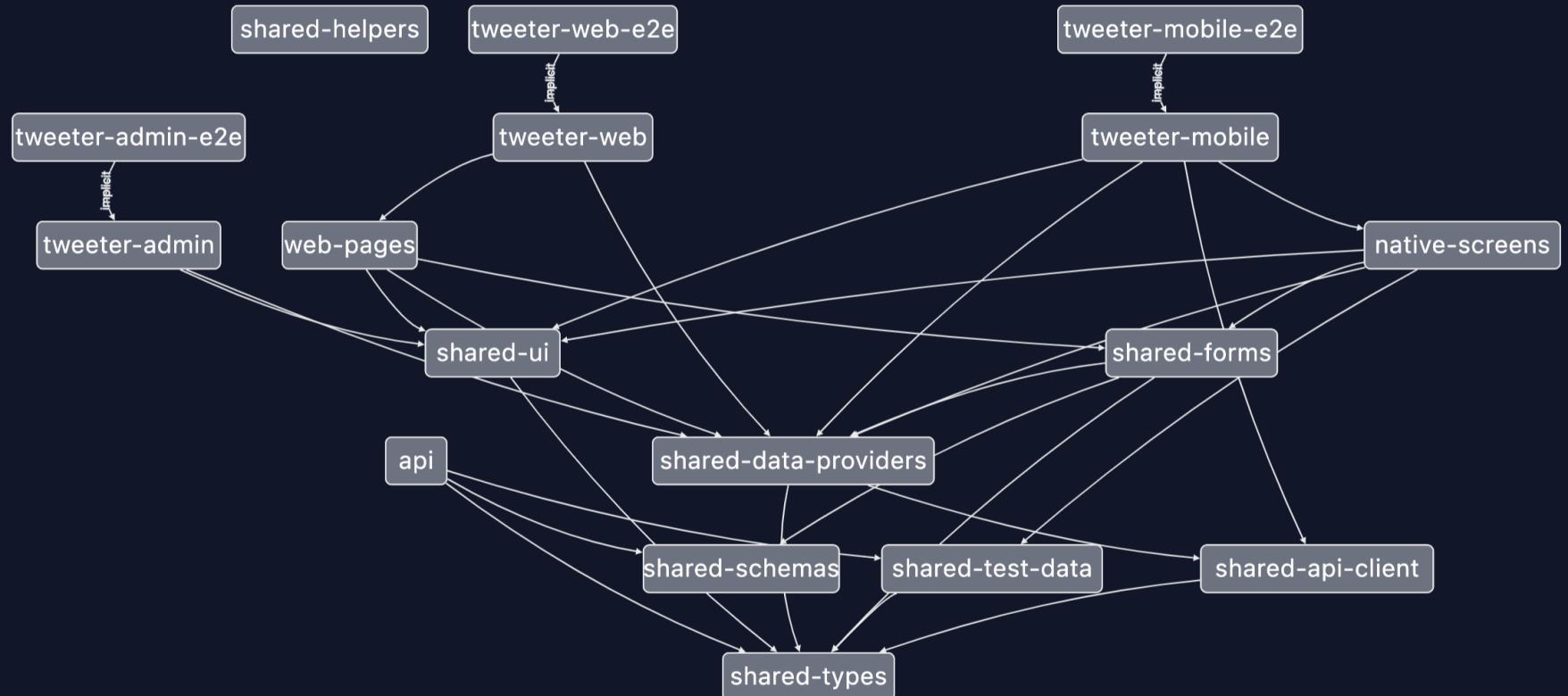
app projects



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 +

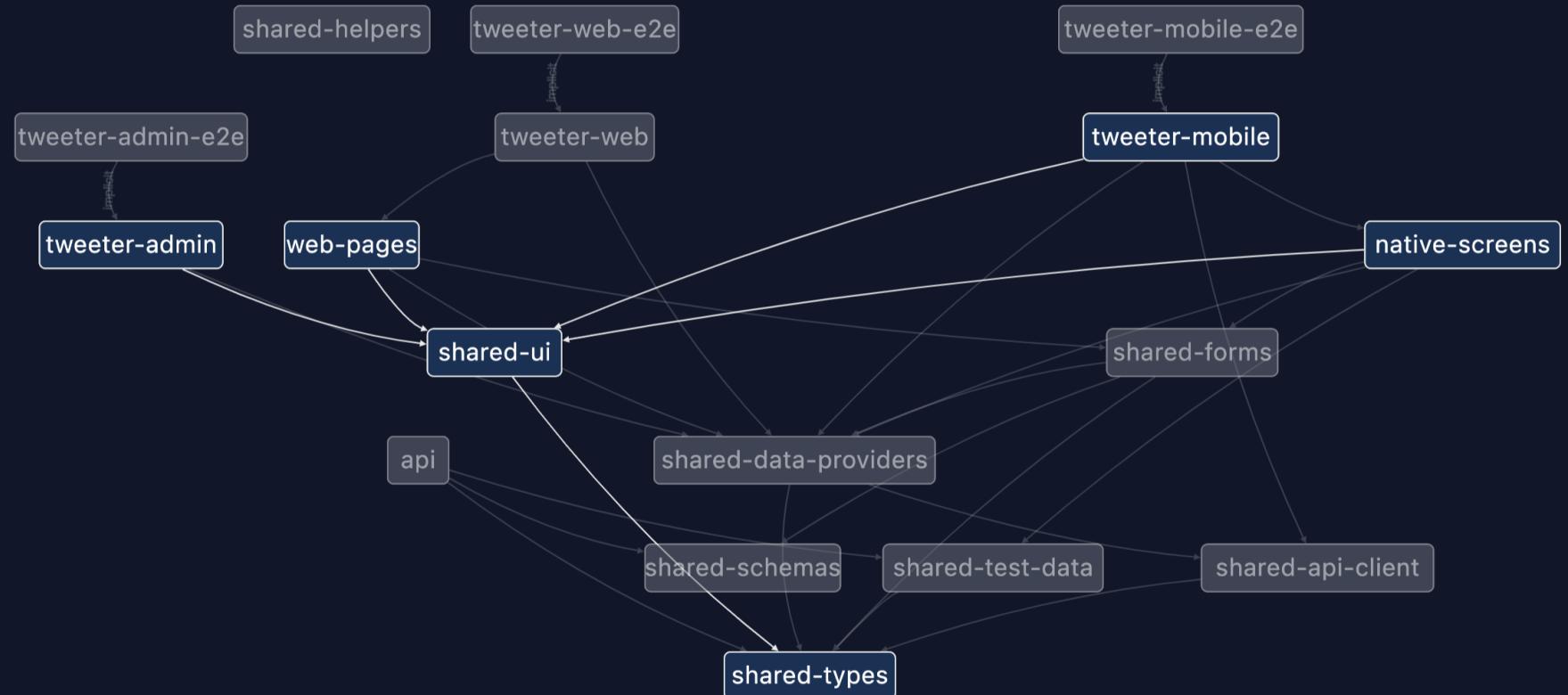
app projects



api



tweeter-admin



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 +

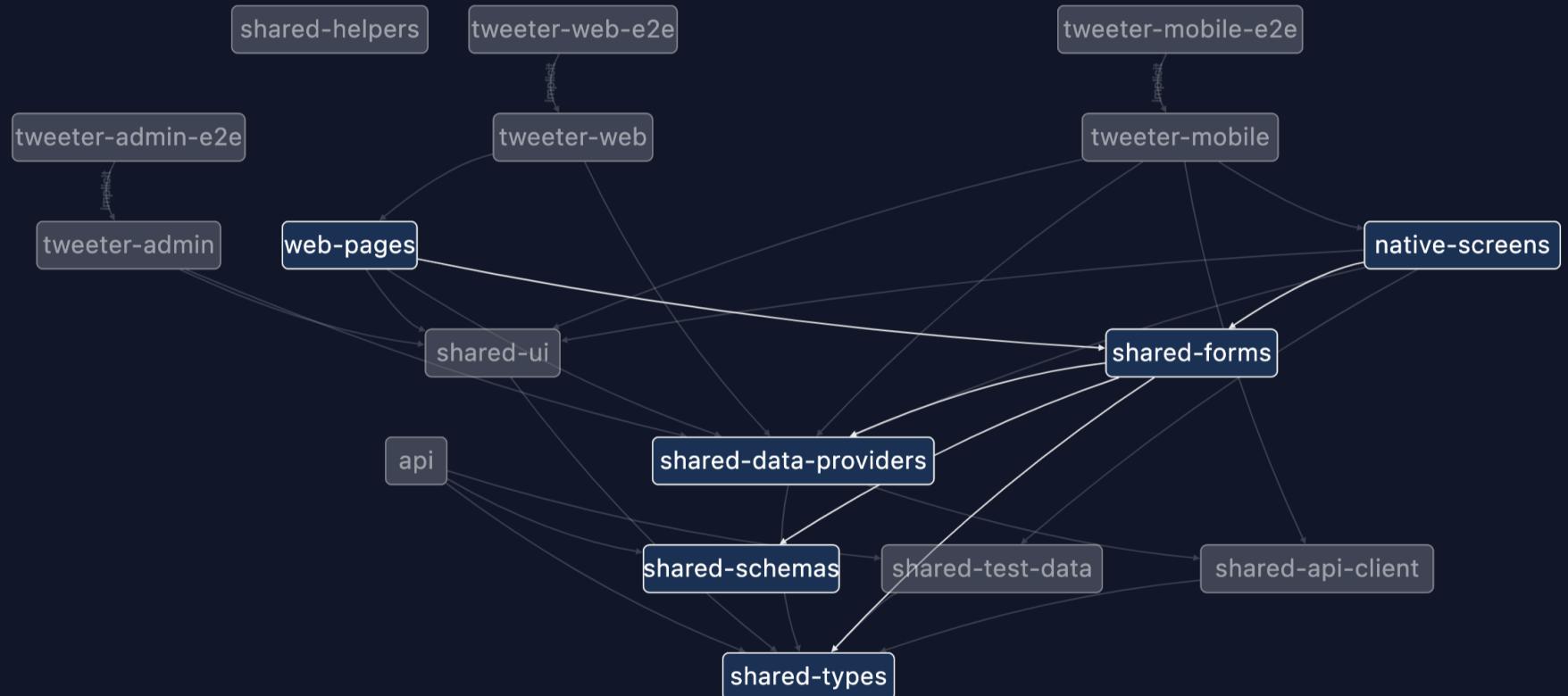
app projects



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 +

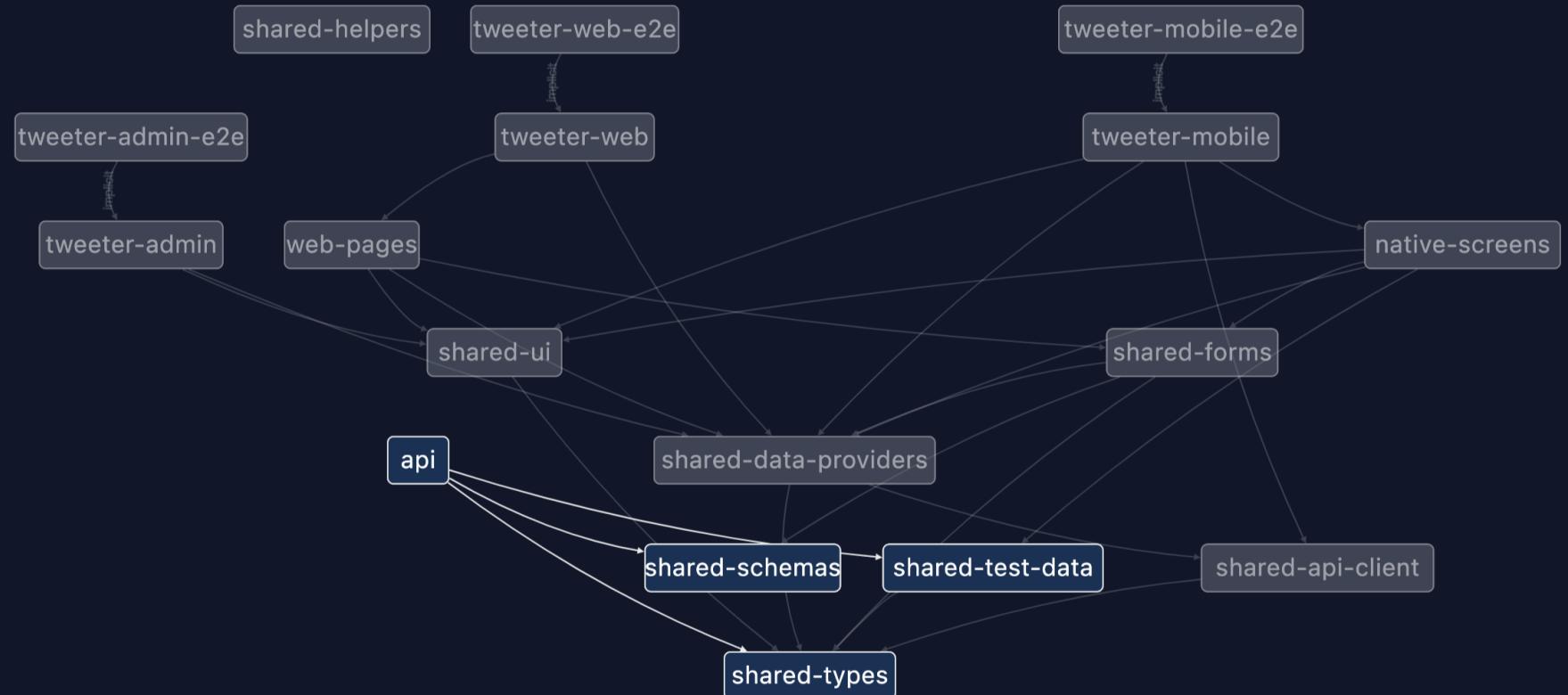
app projects



api



tweeter-admin



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



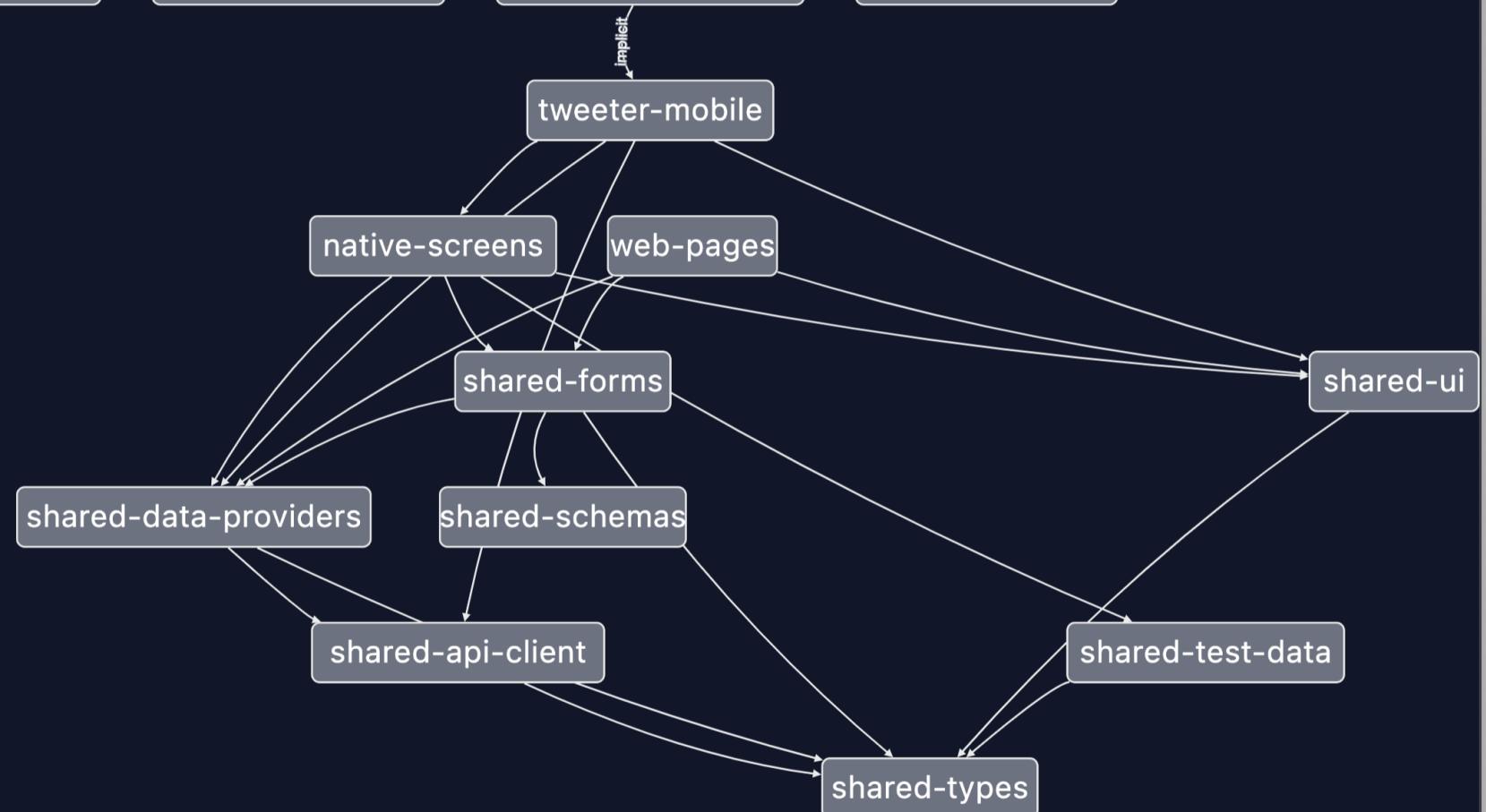
 tweeter-web

shared\helpers

tweeter-admin-e2e

tweeter-mobile-e2e

tweeter-web-e2e



e2e projects

co zrobilbym lepiej?

- z perspektywy czasu zacząłbym od pisania backendu
- rozbicie na środowiska, bo był problem przy pisaniu testów i uruchamianiem aplikacji na środowisku developerskim

- rozbiłbym aplikacje api na biblioteki Nx chociaaz nie mam pomyslu jak 🤔
- przebudowałem aplikacje webowa tak, zeby ekrany mogły byc wspoldzielone miedzy web i admin (web jest zbudowany za pomoca `create-react-app` , admin za pomoca `NextJS`)
- lepsze wspoldzielenie typów

Pytania / pomysły?

Dziękuję za uwagę!

Repo prezentacji i kodu projektu:

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