

Cześć 🙌

Kacper Grzeszczyk

kacgrzes.io / kacgrzes@gmail.com

Jestem programistą JS / TS

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

Byłem trenerem w InfoShare Academy

Smart Front and End-User-Driven Build Systems

Budowanie wieloplatformowych aplikacji za pomocą narzędzia

NX



andre bass
@onlyandrebase

Replying to @elonmusk

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

Elon Musk:



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

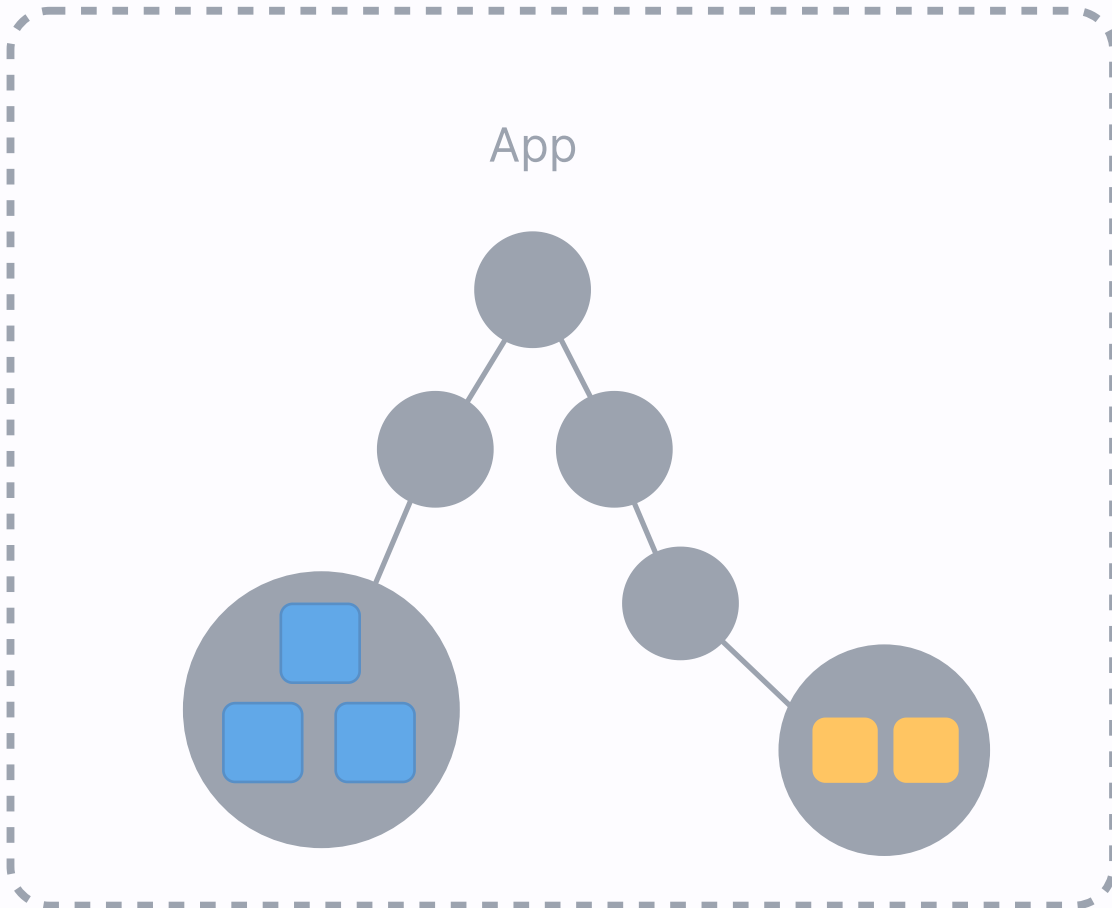
Zbudujemy klon Tittera roboczo nazwany **Twitter** 🐣

Disclaimer

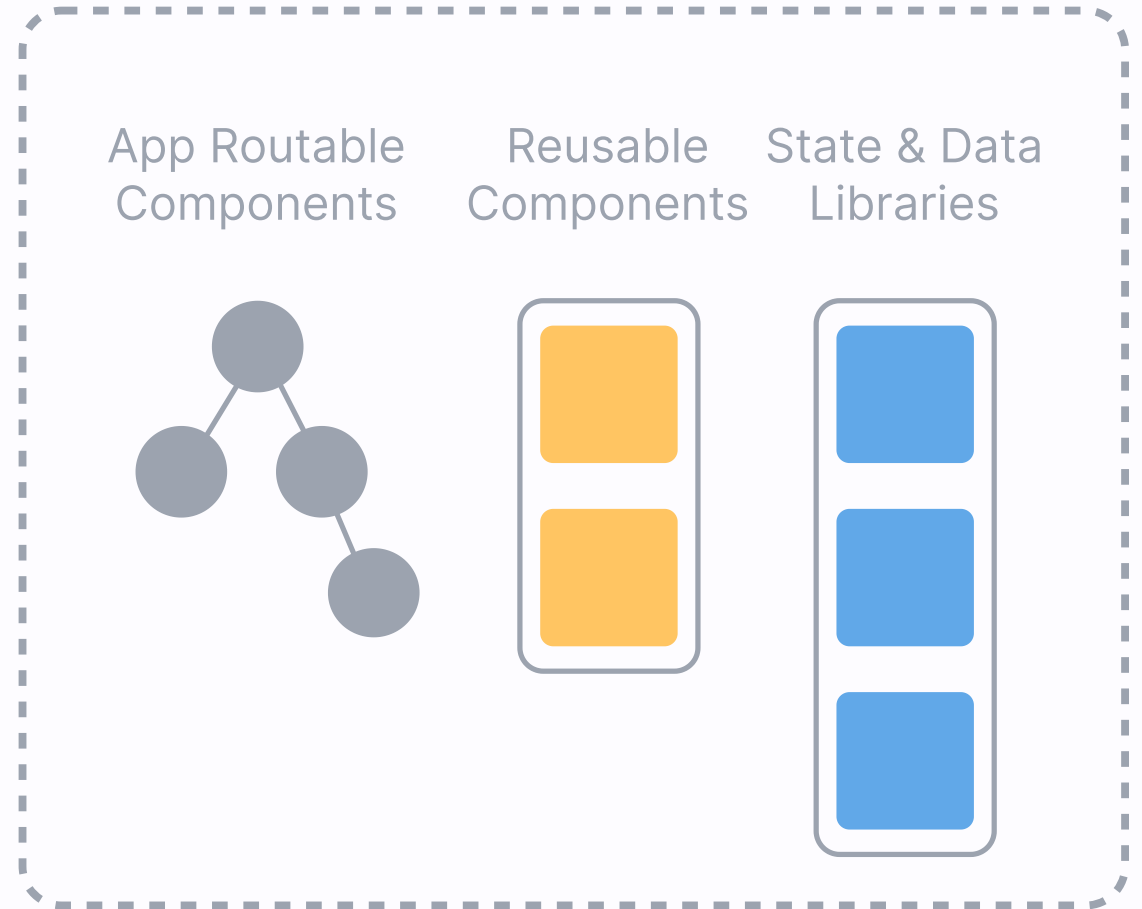
“ Budowanie rozbudowanego klonu na potrzeby prezentacji mija się z celem. Do aplikacji z całą pewnością da się dorzucić więcej funkcjonalności, ale celem prezki jest przede wszystkim pokazanie możliwości budowania projektów na platformie NX. ”

Jak podejść do budowania wieloplatformowej aplikacji? 🤔

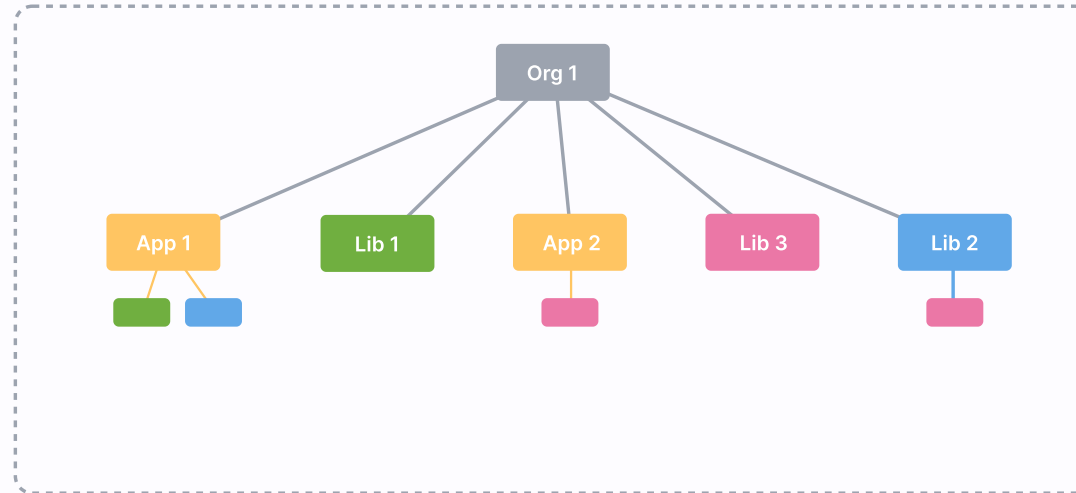
Monolith App Repo



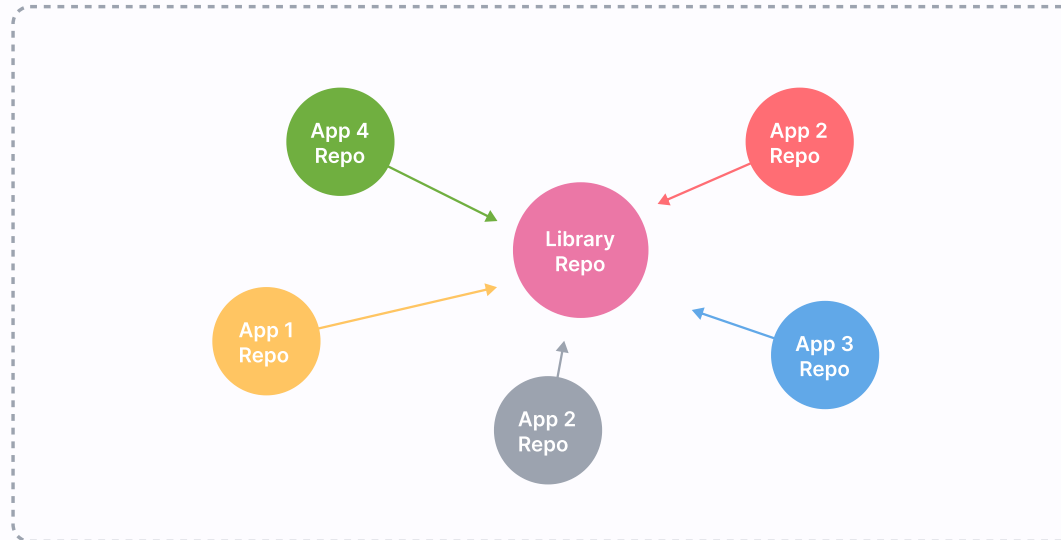
Modular App Repo

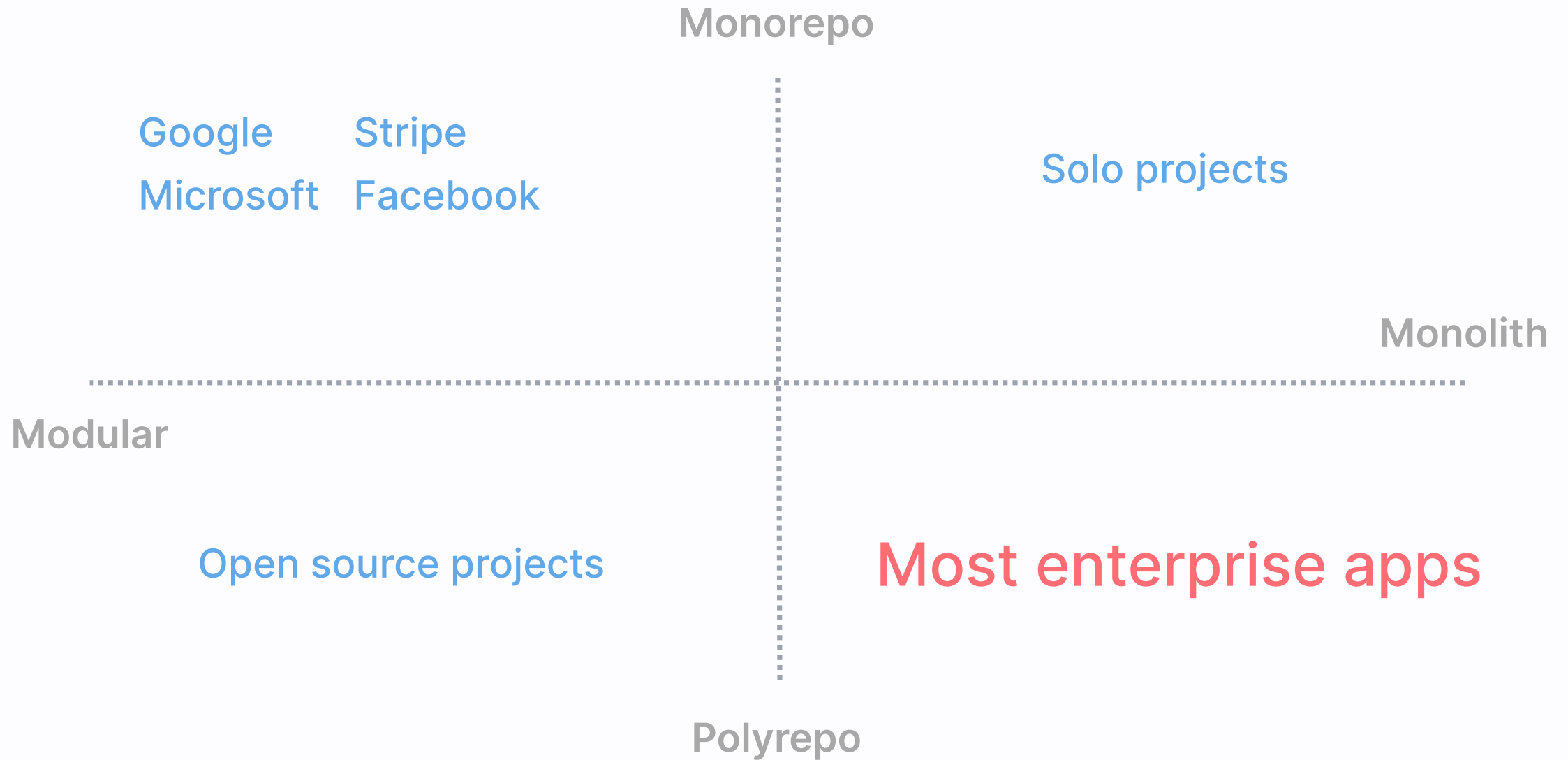



Monorepo



Polyrepo





 więcej na:
<https://monorepo.tools/>

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 moduly 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 ficzery są 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

NX CONSOLE
Nx Generate
☰ ...

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

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


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

```
... TS tweets.router.ts Preview deck.md X Settings {...} settings.json 2
```

```
>nx gener|
```

Nx Console: Focus on **Generate & Run Target View**

Nx: generate

Nx: generate (ui) 

Nx: generate application

Nx: generate library

library|

@nrwl/js - **library** Create a library.

@nrwl/next - **library** Create a library.

@nrwl/node - **library** Create a node library.

@nrwl/react - **library** Create a React library.


@nrwl/react-native - **library** Create a React Native library.

@nrwl/workspace - **library** Create a library.

Search flags

Run

Name *

nx generate @nrwl/js:library 

Directory

Buildable

Compiler

Config

Import Path

Js

Linters

Pascal Case Files

Publishable

Set Parser Options Project

Skip Format

Skip Ts Config

Skip Type Check

Strict

Tags

Test Environment

Unit Test Runner

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

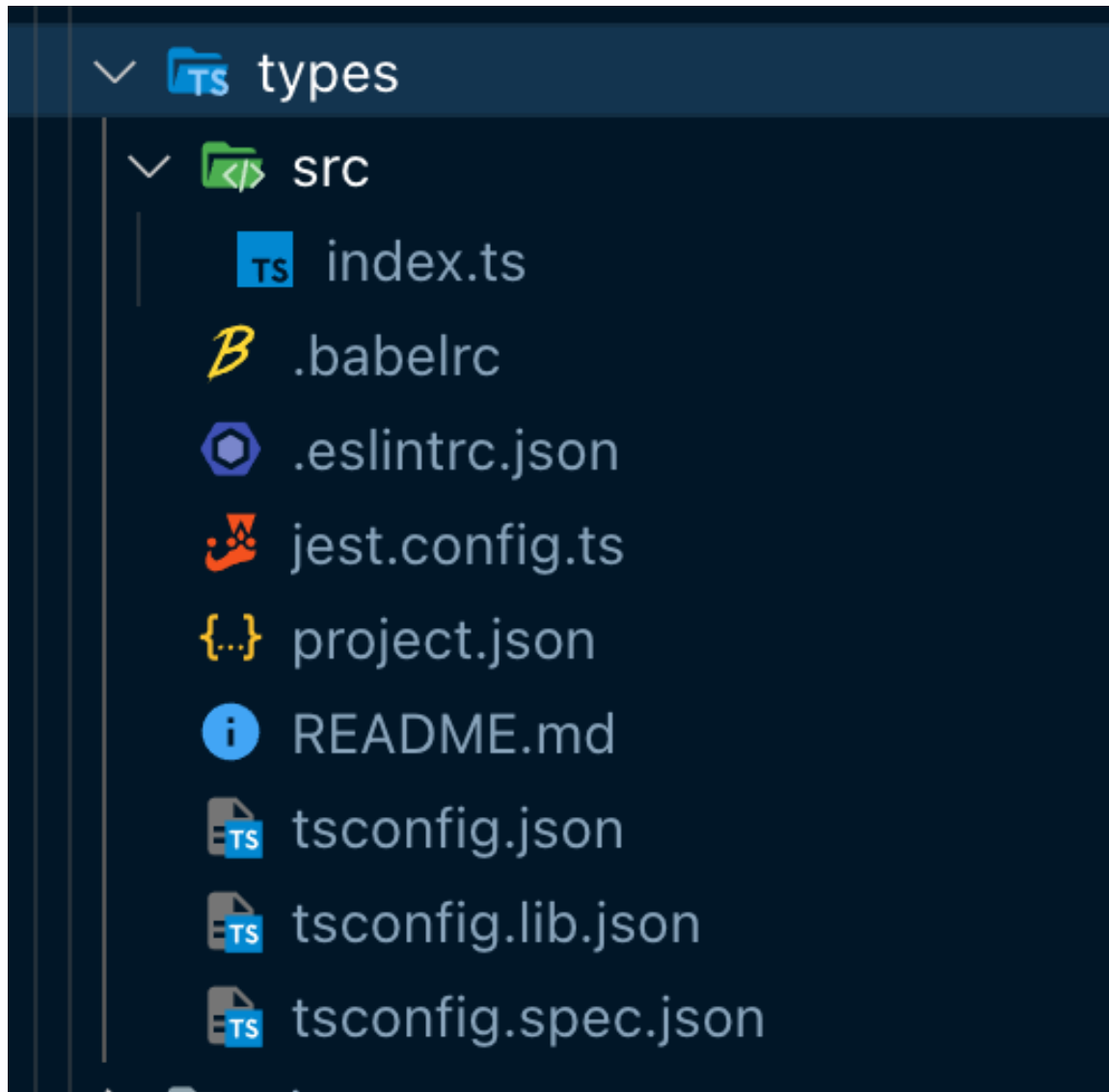
tsc

config

Determines whether the project's executors should be configured in

Ale, dane testowe będą miały określony typ

Więc generujemy kolejną paczkę 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
generującej
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
- Linters
- 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

style

The file extension to be used for style files.

directory

A directory where the lib is placed.

appProject

The application project to add the library route to.

buildable

Generate a buildable library.

ui
--directory=shared

Search flags

Name *

Style

Directory

App Project

Buildable

Compiler

Component

Global Css

Import Path

Js

Linters

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;
};

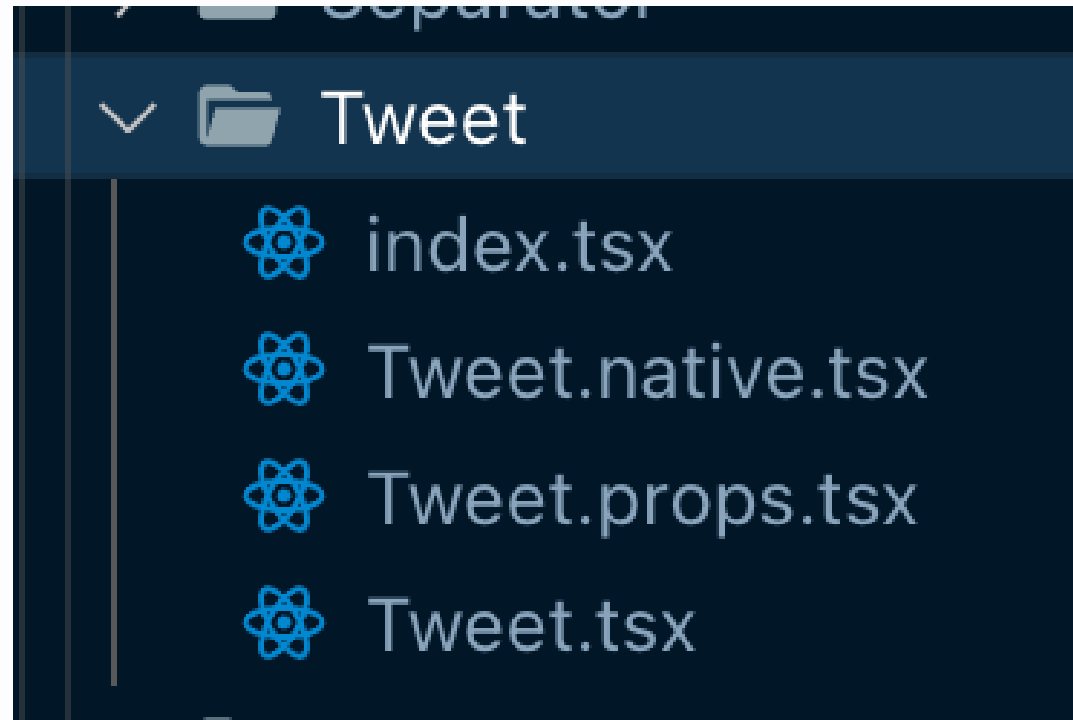
```

```

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 jeśli na webie ma to wyglądać 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}
        />
      </div>
    </div>
  );
};
```

```

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}
        />
      </Pressable>
    </View>
  );
};

```

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={}
/>
```

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

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



@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



@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



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



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



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



qui modi qui magni incidunt 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
 consecetur au



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



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

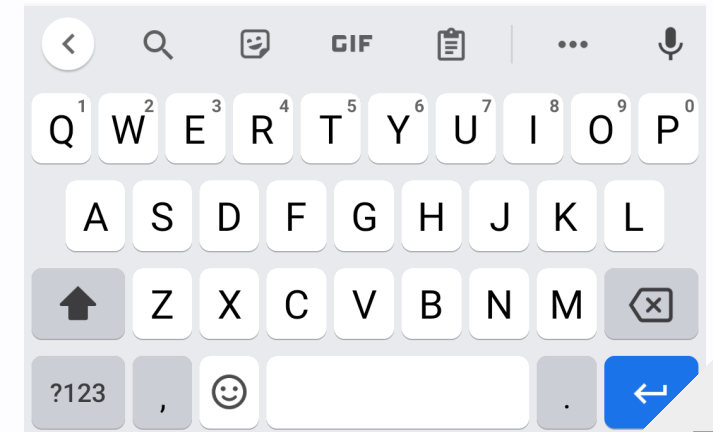
qui modi qui magni incidunt distinctio
quod deserunt magni perferendis
architecto qui eveniet sapiente voluptate
facilis similique tenetur est esse
explicabo si

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

Napisałem w `express.js` za pomocą metodyki TDD (tutaj chciałem być pewny, że wszystko działa jak należy)

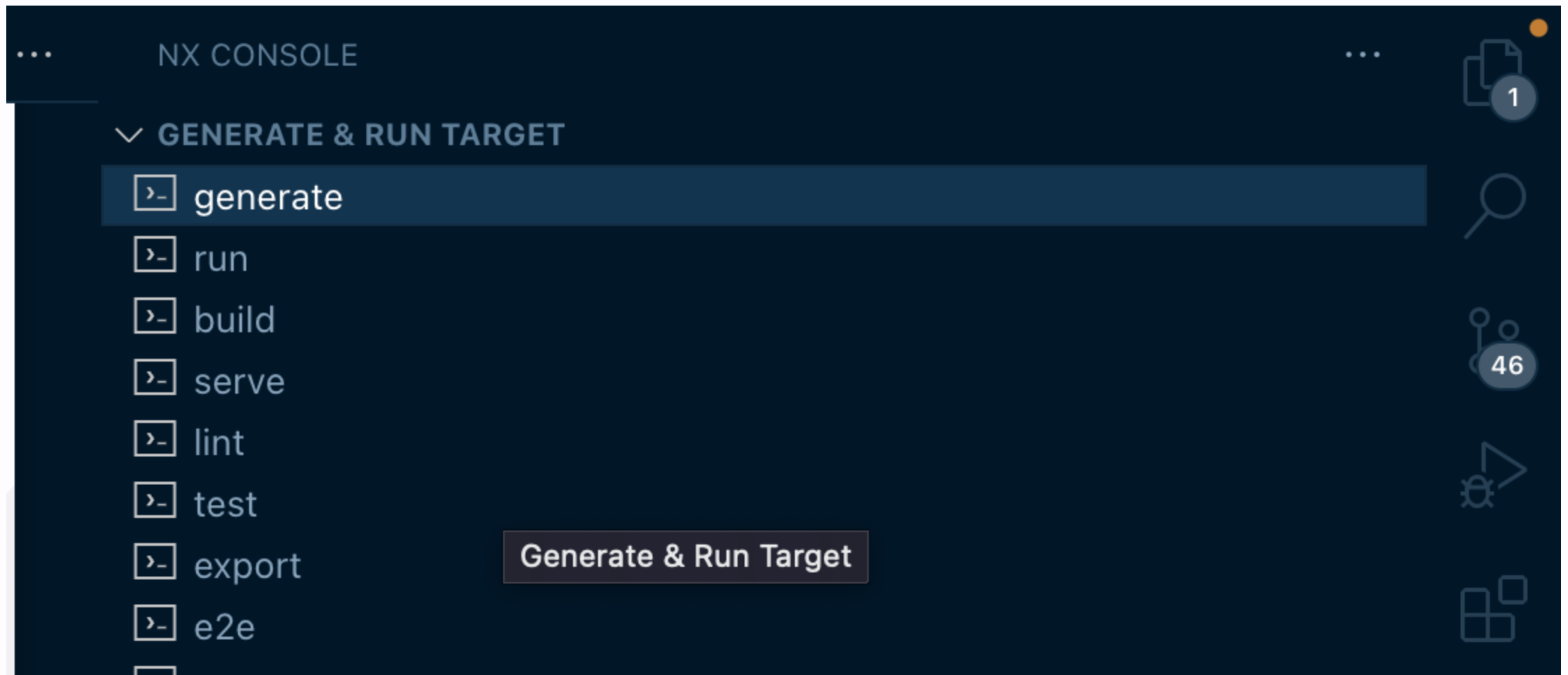
Instaluję kolejny generator:

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

i generuję apke za pomocą CLI 

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

albo wyklikuję z UI 



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

name

The name of the application.

api

api

directory

The directory of the new application.

babelJest

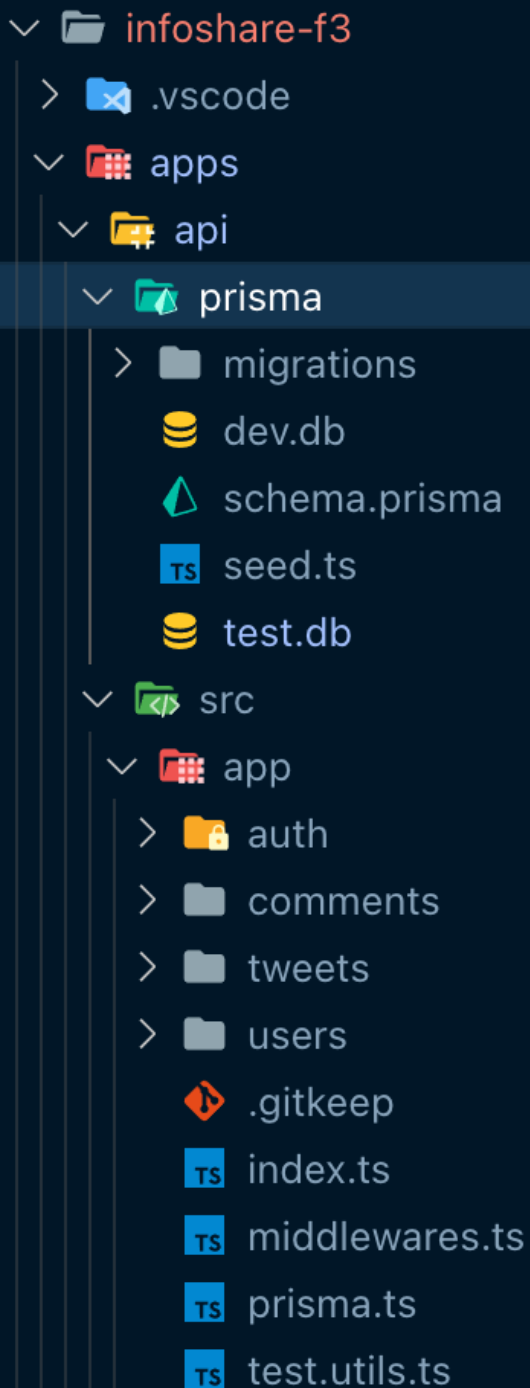
Use `babel` instead `ts-jest`.

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 napisałem 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);
    }
  },
});
```

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

Kończenie aplikacji mobilnej

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

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

Aplikacja webowa

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

Browser window showing a Twitter-like interface on localhost:4200. The page displays a list of tweets with user avatars, names, timestamps, and text content. The tweets are:

- Co się dzieje?** (No user name visible)
- @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 replies, 29 likes
- @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 replies, 30 likes
- @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 replies, 30 likes
- @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
14 replies, 30 likes

A "Tweet" button is visible in the top right of the first tweet's input area.

TweeterWeb

localhost:4200/tweets/cl38kchw8002k5ncvgnl0vfr

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 administratora


Welcome to tweeter-admin! x +

localhost:4300/users Guest

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


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

Delete tweet

 @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

Delete tweet

 @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


Delete tweet

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

14 26

Delete tweet

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

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

Co można było jeszcze wspoldzielić poza częściami UI?

- między wszystkimi aplikacjami - typy, helpery, schemas
- między 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);

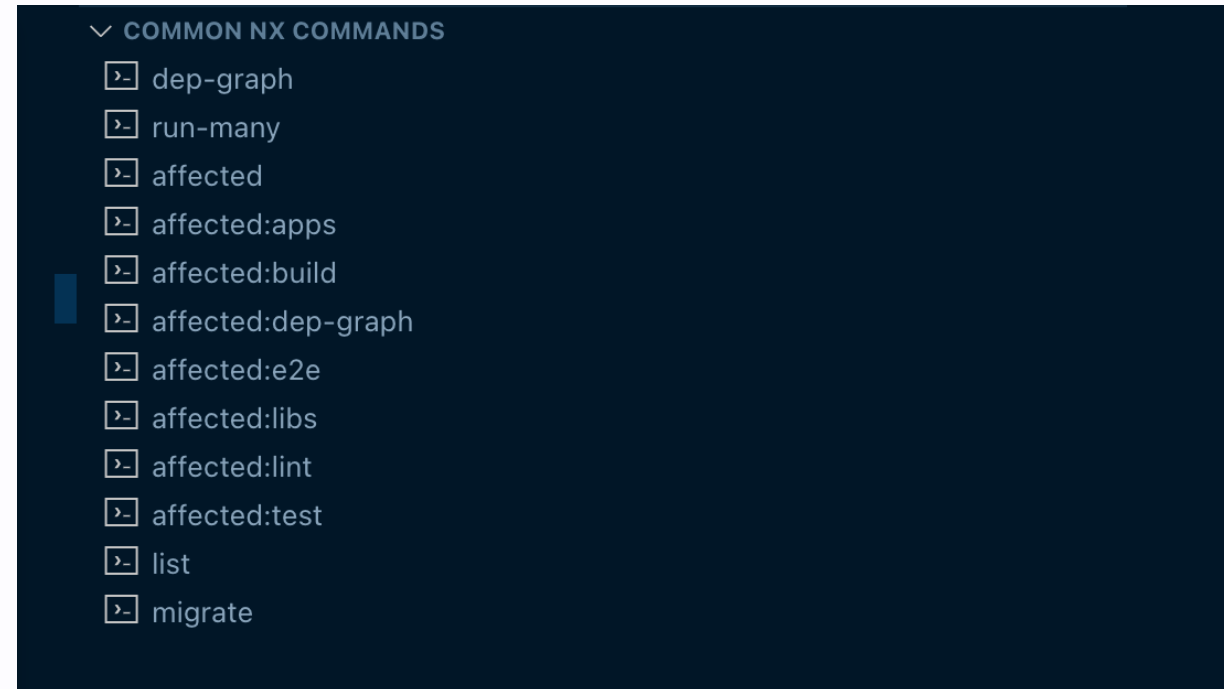
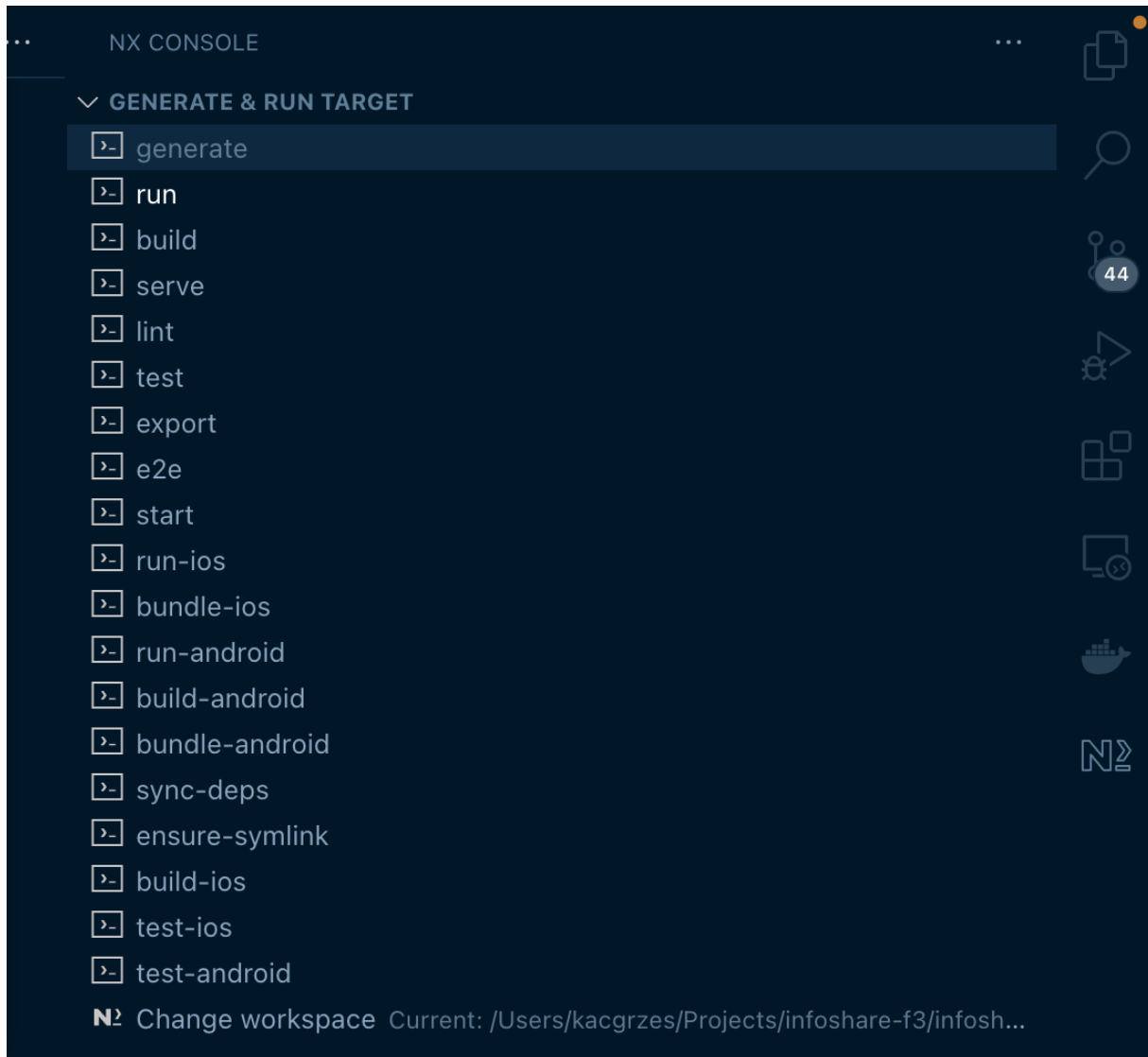
  return {
    ...createTweetForm,
    onSubmit,
  };
};

```

Reguła którą 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

Execute command or set flags

Execute: `nx run-many --target=serve`

projects

Projects to run

all

All projects

parallel

Parallelize the command

maxParallel

Max number of parallel processes

only-failed

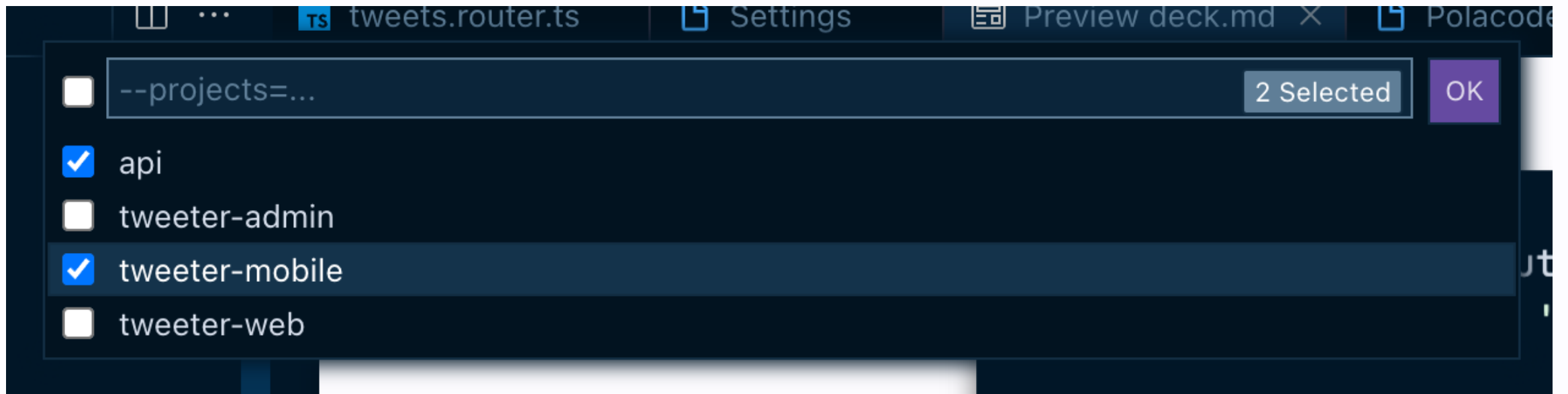
Isolate projects which previously failed

configuration

This is the configuration to use when performing tasks on projects

skip-nx-cache

Rerun the tasks even when the results are available in the cache



Generowanie grafów zależności

caty graf

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

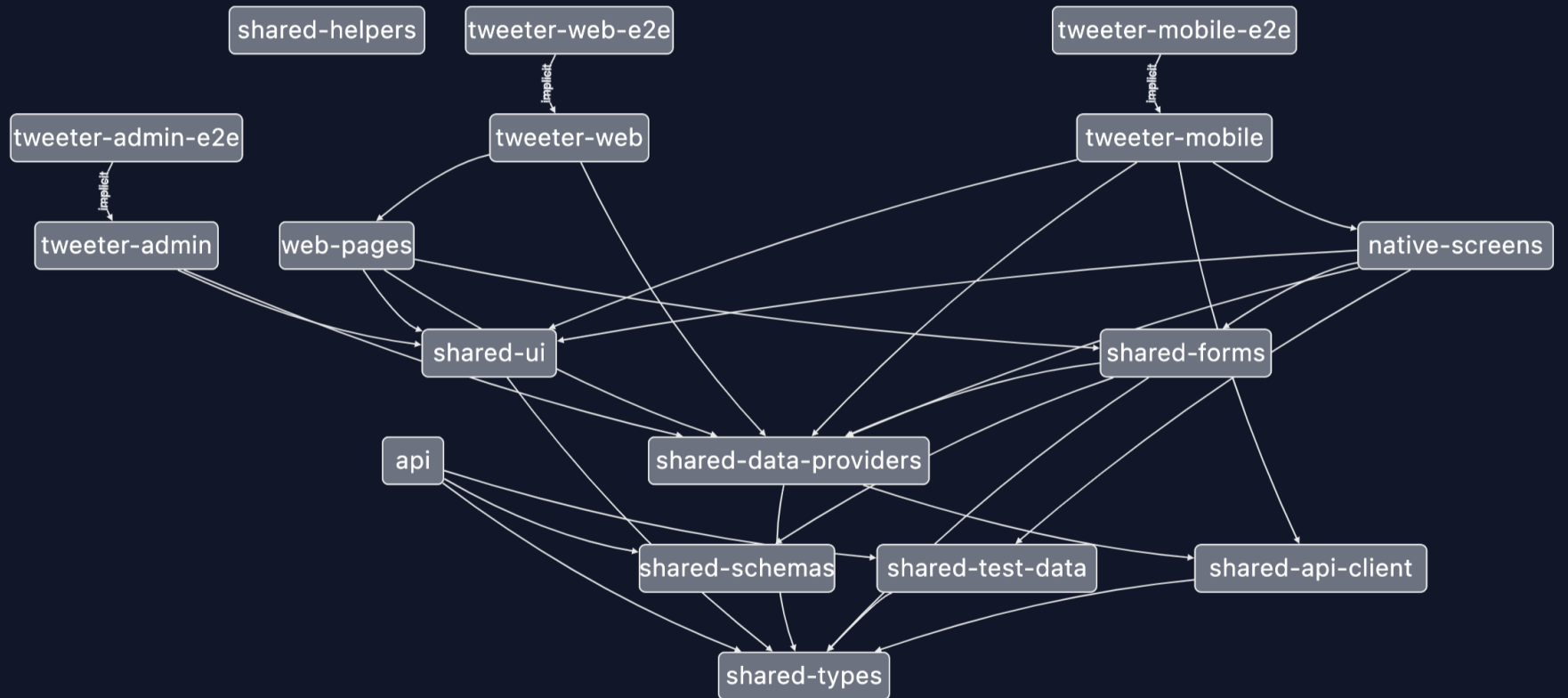
Group by folder
Visually arrange libraries by folders with different colors.

Activate proximity
Explore connected libraries step by step.

— 1 +

app projects

-  api 
-  tweeter-admin 



zaleznosci apki mobilnej

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.

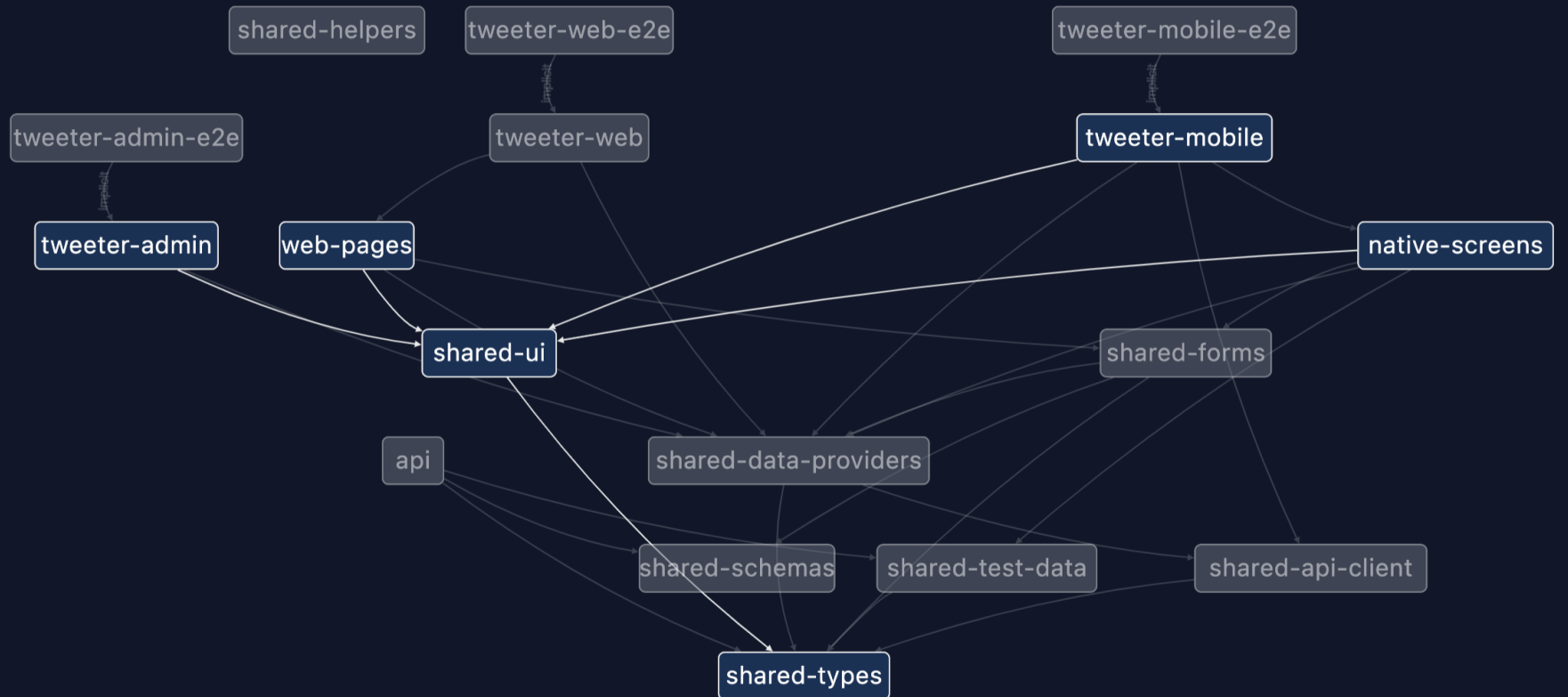
Activate proximity
Explore connected libraries step by step.

— 1 +

app projects

📁 api **👁**

📁 tweeter-admin **👁**



wspoldzielone formularze

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.

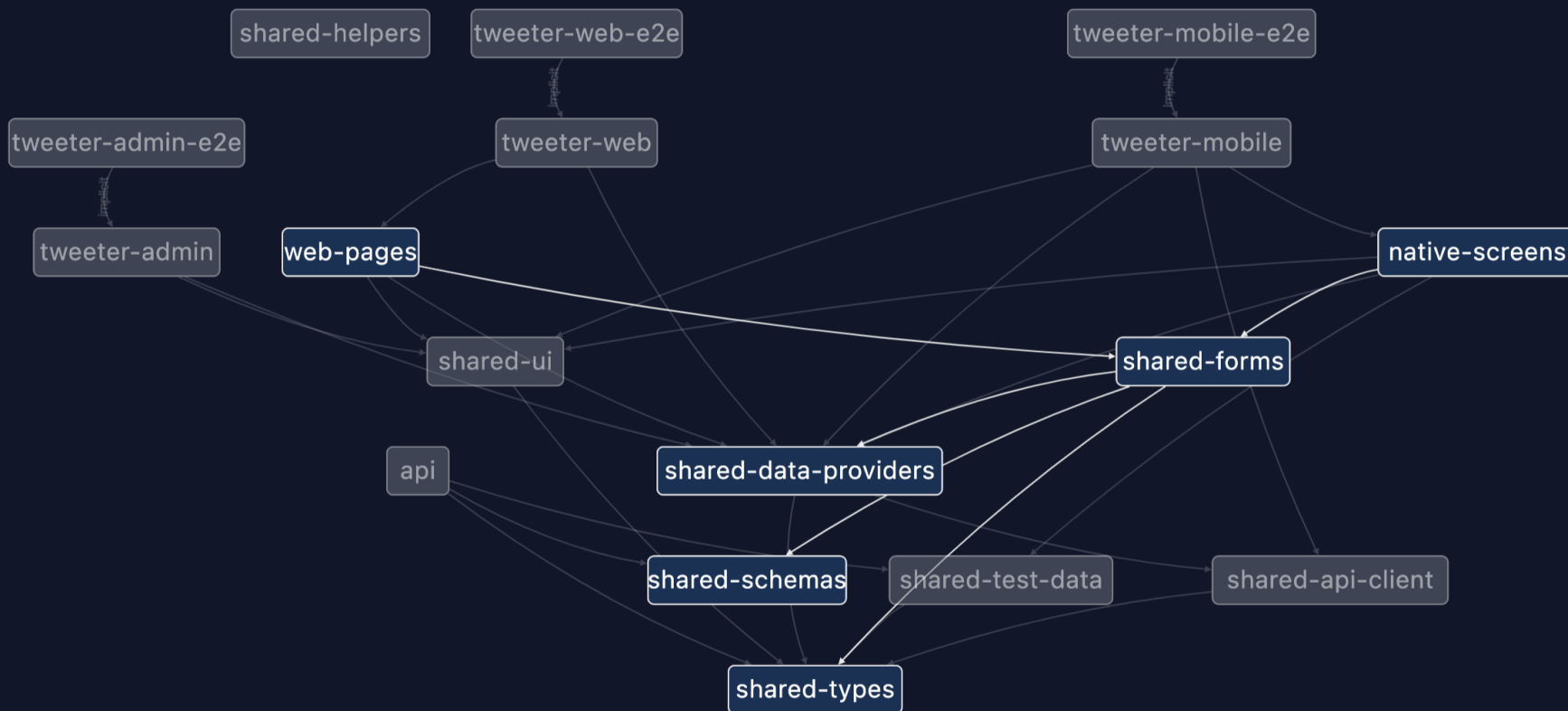
Activate proximity
Explore connected libraries step by step.

— 1 +

app projects

📁 api **👁**

📁 tweeter-admin **👁**



zaleznosci api

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

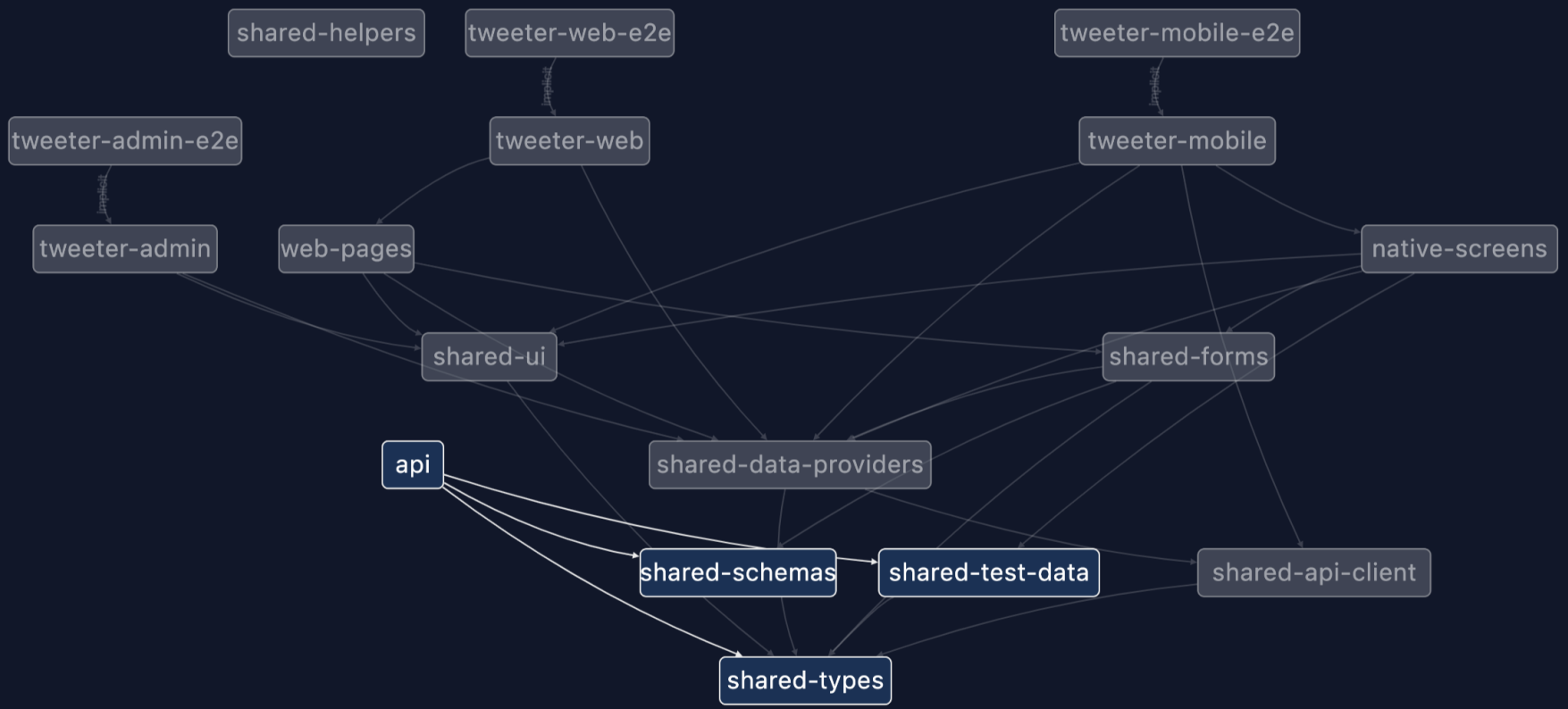
Group by folder
Visually arrange libraries by folders with different colors.

Activate proximity
Explore connected libraries step by step.

— 1 +

app projects

-  api 
-  tweeter-admin 



odfiltrowana apka mobilna

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.

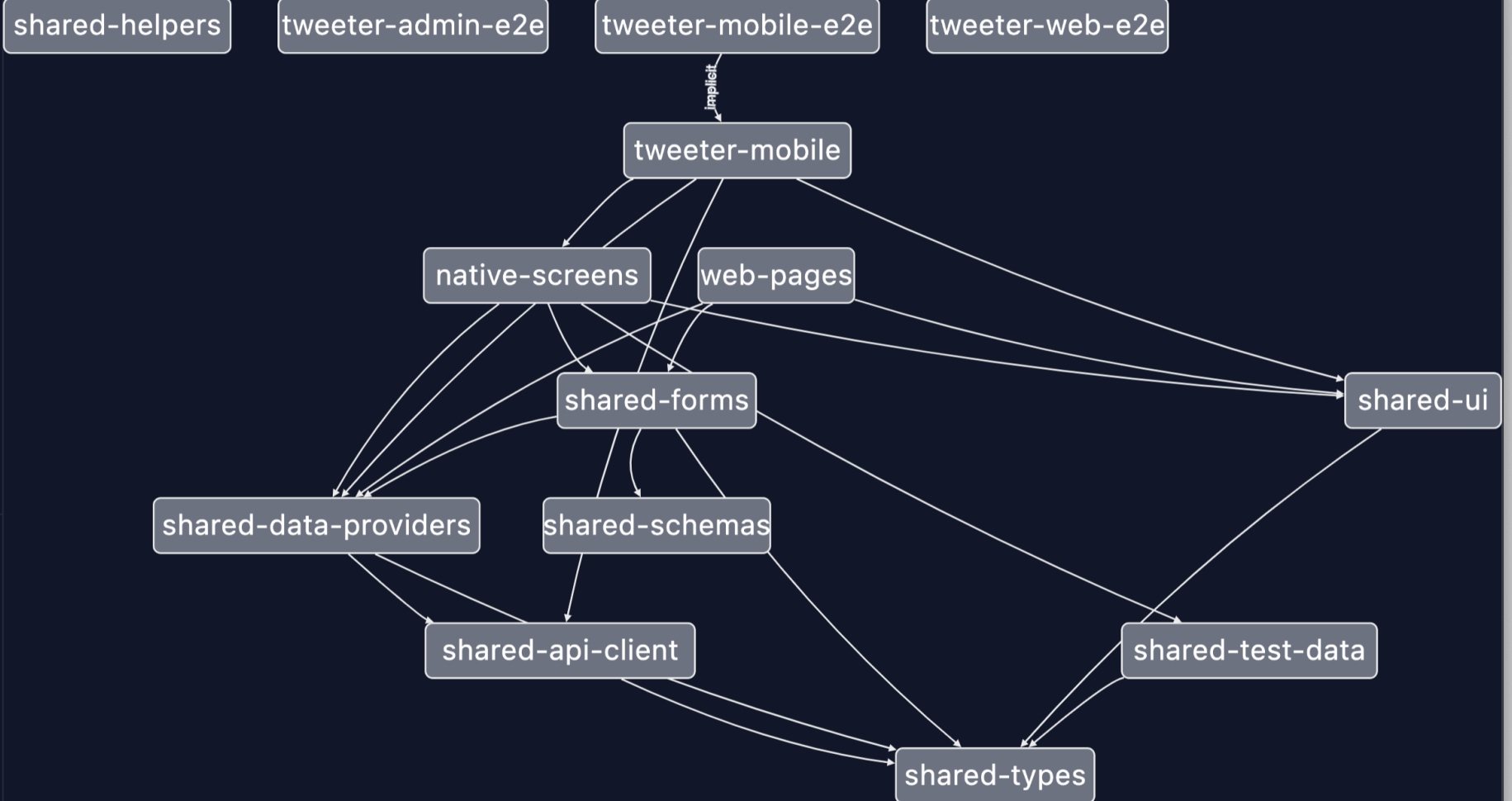
Activate proximity
Explore connected libraries step by step.

— 1 +

app projects

- api
- tweeter-admin
- tweeter-mobile
- tweeter-web

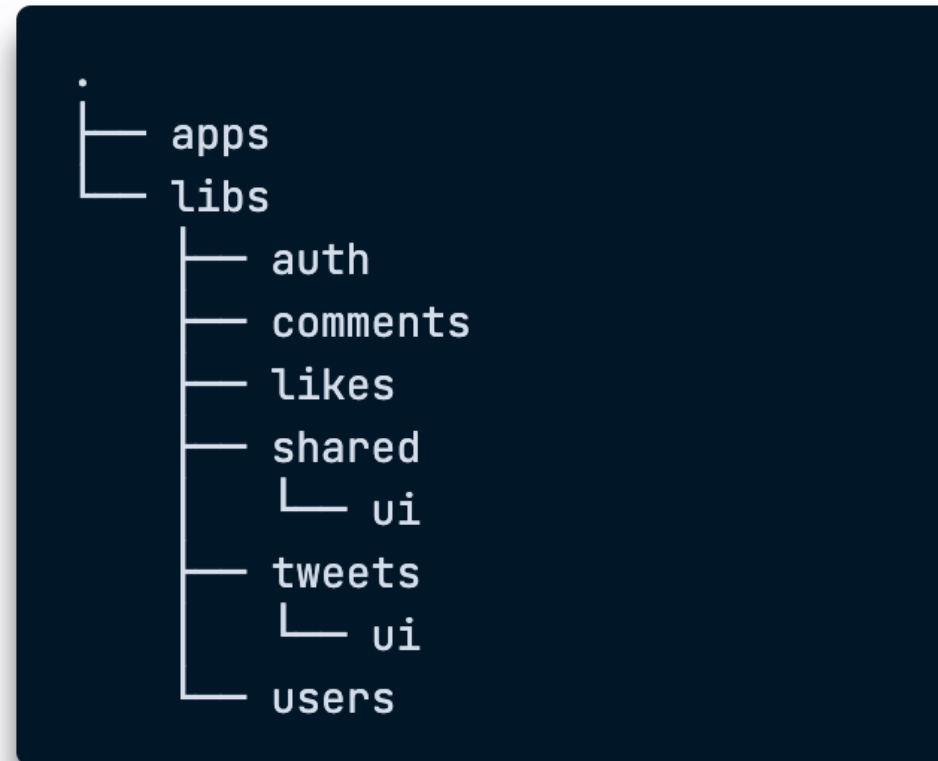
e2e projects



co zrobiłbym lepiej?

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

- rozbiłbym aplikacje api na biblioteki i dzieliłbym kod domenowo



- przebudowałbym aplikacje webowa tak, zeby ekrany mogly byc wspoldzielone miedzy web i admin (web jest zbudowany za pomoca `create-react-app` , admin za pomoca `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