

TypeScript

ANGULAR

Instalacja -g oznacza zainstalowanie angulara globalnie czyli będzie można go używać w każdym projekcie który strzowycie w Node.js

```
npm install -g @angular/cli
```

Generowanie nowe projektu: (sprawdź gdzie aktualnie się znajdujesz w konsoli – polecenie: pwd)

```
ng new NAZWA_PROJEKTU
```

Domyślnie używany jest CSS ale można podczas generowania projektu podać jaki „język” stylu chcesz użyć: np.:

```
ng new NAZWA_PROJEKTU --style=scss
```

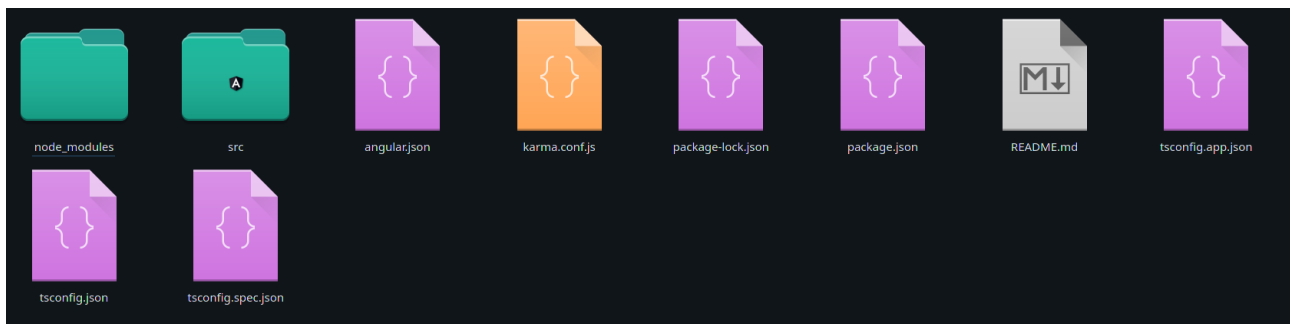
Stworzenie nowego projektu wygeneruje w konsoli coś podobnego do poniższego:

```
adams #>: ng new ANGULAR_1
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? CSS
CREATE ANGULAR_1/README.md (1062 bytes)
CREATE ANGULAR_1/.editorconfig (274 bytes)
CREATE ANGULAR_1/.gitignore (548 bytes)

v Packages installed successfully.
podpowiedź: Użycie „master” jako nawy początkowej gałęzi. Ta domyślna nazwa gałęzi
podpowiedź: może się zmienić. Aby ustawić nazwę początkowej gałęzi do użycia we
podpowiedź: wszystkich nowych repozytoriach, co uciszy ten komunikat, wykonaj:
podpowiedź: git config --global init.defaultBranch <nazwa>
podpowiedź: Tradycyjnie było „master”, a inne popularne to „main”, „trunk”
podpowiedź: i „development”. Nazwę właśnie utworzonej gałęzi można zmienić tym poleceniem:
podpowiedź: git branch -m <nazwa>
Successfully initialized git.
```

Jako że mam zainstalowanego GIT to pojawiły mi się powyższe podpowiedzi.

W folderze wygenerują się następujące pliki i foldery:



Już można uruchomić naszą aplikację. Wchodzimy do utworzonego projektu poleceniem `ng new NAZWA_PROJEKTU` – u mnie nazwa to **ANGULAR_1**. Pod linuxem wykonujesz polecenie `cd ANGULAR_1`

Następnie uruchamiamy naszą aplikację poleceniem:

`ng serve`

Chwilę należy poczekać. W terminalu pojawi się:

```
adams #>: ng serve

? Would you like to share pseudonymous usage data about this project with the Angular Team
at Google under Google's Privacy Policy at https://policies.google.com/privacy. For more
details and how to change this setting, see https://angular.io/analytics. No
Global setting: enabled
Local setting: disabled
Effective status: disabled
✔ Browser application bundle generation complete.

Initial Chunk Files | Names          | Raw Size
vendor.js           | vendor         | 2.11 MB
polyfills.js        | polyfills      | 318.01 kB
styles.css, styles.js | styles        | 210.09 kB
main.js             | main           | 49.84 kB
runtime.js          | runtime        | 6.52 kB

| Initial Total | 2.68 MB

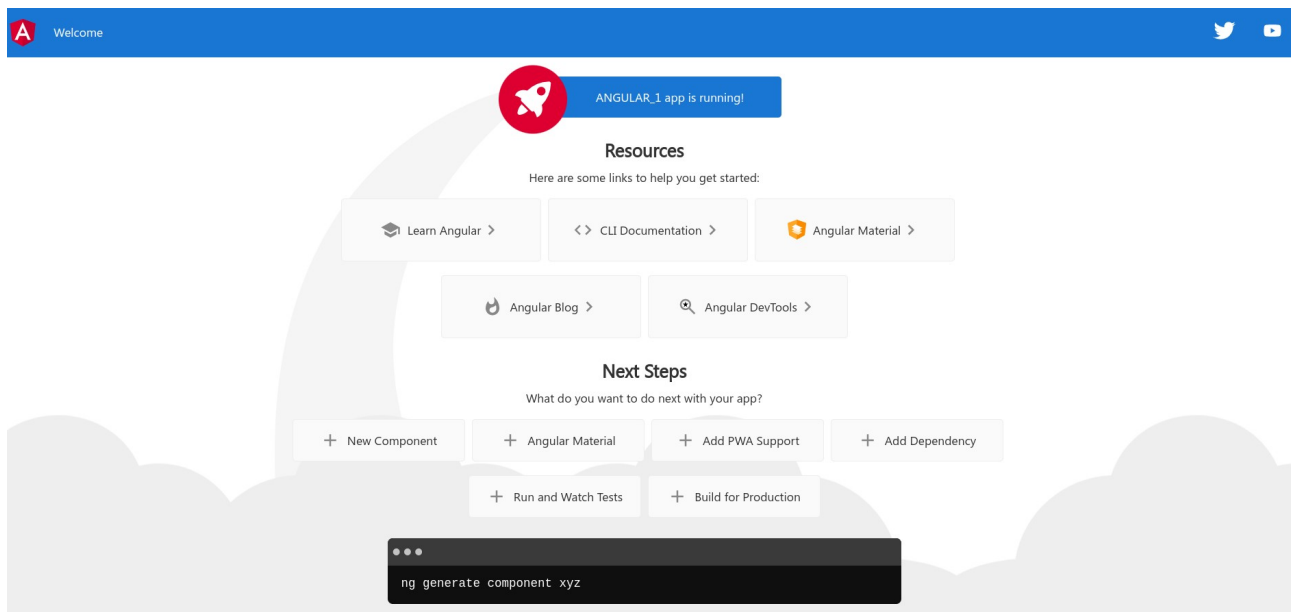
Build at: 2023-03-06T10:02:36.389Z - Hash: 31c82587df793dfb - Time: 9984ms

** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

✔ Compiled successfully.
```

Teraz możemy w przeglądarce wpisać: <http://localhost:4200> pojawi się nam strona:

ZWRÓĆ UWAGĘ NA PORT → 4200



Teraz postaram się opisać co znajduje się w poszczególnych plikach:
Na samym dole znajdziecie pliki konfiguracyjne naszego projektu.

.editorconfig – konfiguracja Visual Studio Code

.gitignore – mówi o tym, które pliki nie powinny wylądować w repozytorium.

Po podpięciu repozytorium pojawi się również folder **.git**.

package.json – zbiór wszystkich paczek, zależności wykorzystywanych w projekcie.

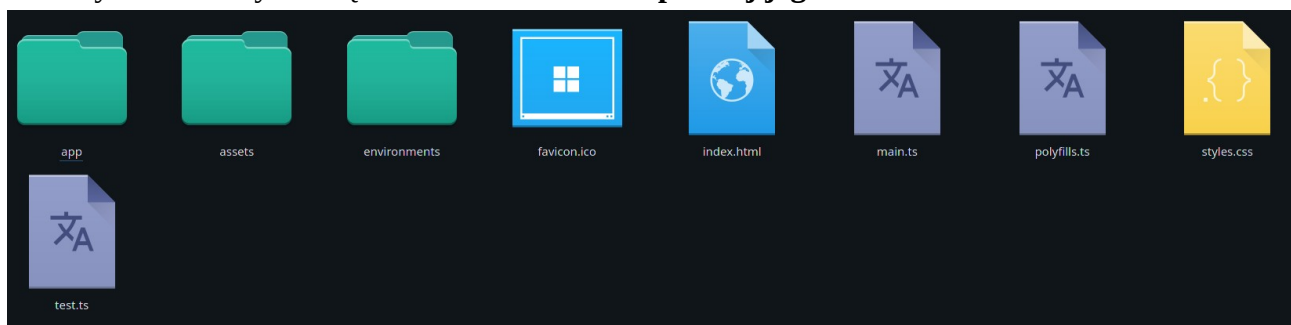
node_modules – moduły zainstalowane w naszym projekcie.

Pliki konfiguracyjne projektu TypeScript:

tsconfig.json

tsconfig.app.json oraz tsconfig.spec.json

Główny folder który nas będzie interesował: **src** – **poniżej jego zawartość**



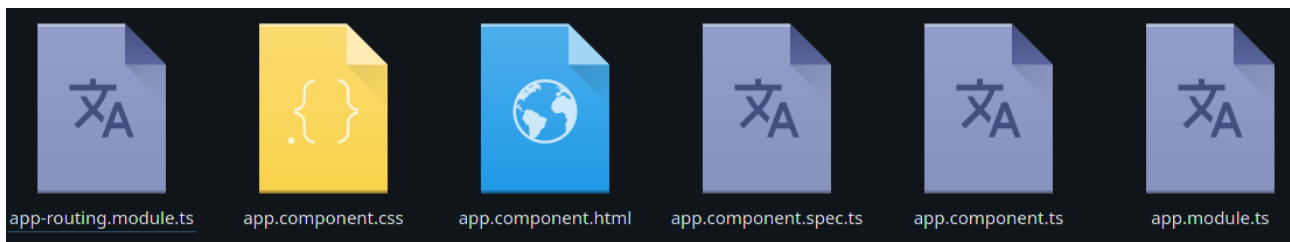
Pliki główne:

index.html

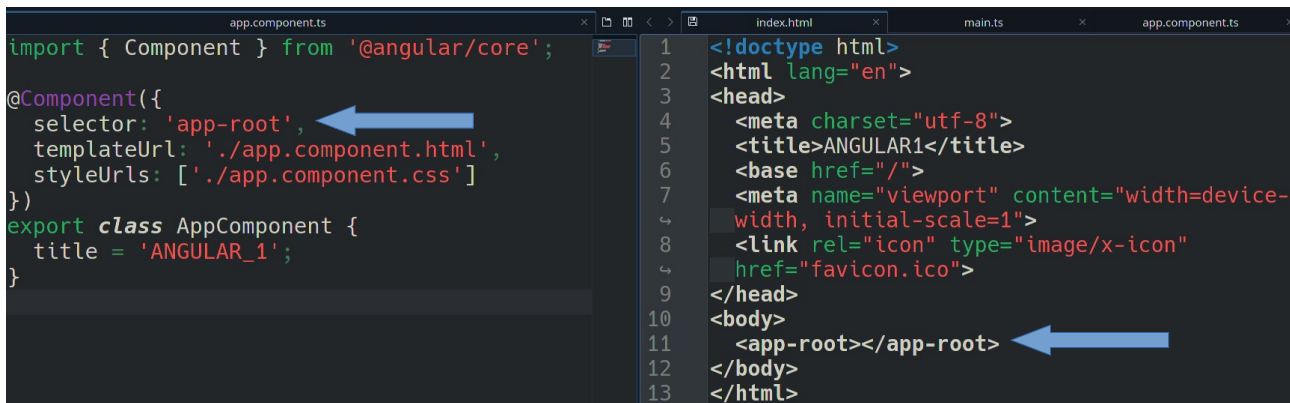
main.ts – tutaj startuje nasza aplikacja

styles.css

Cała aplikacja znajduje się w folderze **app**.

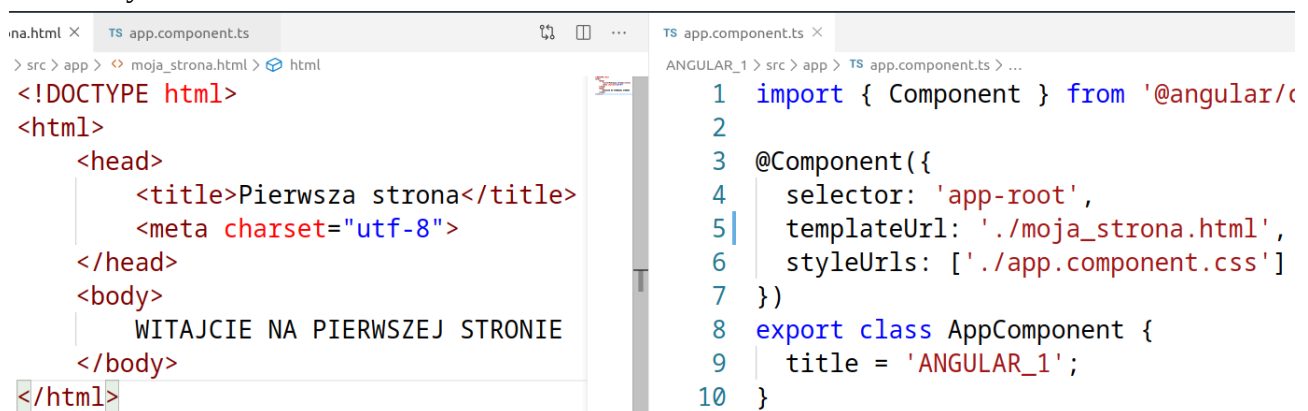


Konfiguracja aplikacji znajduje się w pliku **app.component.ts** tam są odwołania do pliku głównego **index.html** (app-root) odwołanie do naszej strony: **templateUrl** oraz odwołanie do stylów: **styleUrls**:



templateUrl – określa jaki plik używamy do naszej strony internetowej czyli bazowy plik HTML.

Po drobnych zmianach:



Strona wygląda jak poniżej:


WITAJCIE NA PIERWSZEJ STRONIE

Mechanizm interpolacji:

app.components.ts

```
...
})
export class AppComponent {
  title = 'WITAJCIE';
  powitanie:string = "Wyświetla się dla
}

```

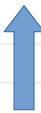


moja_strona.html

```

7    <body>
8      <h4>WITAJCIE
9      {{powitanie}}
10   </body>
11 </html>

```



Konieczniew podwójnych nawiasach kłamrowych

Na stronie pojawi się (kolor dla WITAJCI ... zmienił się bo zmieniono kolor w CSS:

WITAJCIE NA PIERWSZEJ STRONIE

Wyświetla się dlatego że mechanizm ten nazywa się interpolacją

3adania

Stwórz projekt w Node.js z wykorzystaniem Angulara. Na stronie powinien znaleźć się napis sformatowany za pomocą CSS:

Autor: Jan Kowalski //Oczywiście podajesz swoje dane.

{{data_utworzenia}}

{{data_utworzenia}} – zastosowanie mechanizmu interpolacji gdzie data wpisana jest na stałe