Wydanie III

Angular

Dziesięć praktycznych aplikacji internetowych z wykorzystaniem najnowszych rozwiązań technologicznych

Aristeidis Bampakos

Helion

‹packt›

Tytuł oryginału: Angular Projects Build modern web apps in Angular 16
with 10 different projects and cutting-edge technologies, 3rd Edition

Tłumaczenie: Krzysztof Bąbol

ISBN: 978-83-289-0809-3

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
https://helion.pl/user/opinie/angdz3
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Helion S.A.
ul. Kościuszki 1c, 44-100 Gliwice
tel. 32 230 98 63
e-mail: *helion@helion.pl*
WWW: *https://helion.pl* (księgarnia internetowa, katalog książek)

Printed in Poland.

Spis treści

Tworzenie pierwszej aplikacji internetowej w Angularze

Rozdział 1

Angular jest popularnym i nowoczesnym frameworkiem języka **JavaScript** działającym na różnych platformach, między innymi w sieci WWW, na komputerach i na urządzeniach mobilnych. Aplikacje Angulara pisane są w języku **TypeScript**, który jest nadzbiorem JavaScriptu z lukrem składniowym zapewniającym między innymi silne typowanie i możliwość stosowania technik obiektowych.

Aplikacje Angulara tworzy się i opracowuje przy użyciu dostarczonego przez jego zespół narzędzia wiersza poleceń o nazwie **Angular CLI**. Automatyzuje ono wiele zadań projektowych, między innymi tworzenie rusztowań (ang. *scaffolding*), testowanie i wdrażanie aplikacji, co przy ręcznej konfiguracji zajmowałoby wiele czasu.

Popularność tego frameworka znajduje w znacznym stopniu odzwierciedlenie w szerokim wsparciu narzędziowym. Wiele rozszerzeń, które ułatwiają programistom pracę z Angularem, zawiera edytor **Visual Studio Code** (**VS Code**).

W tym rozdziale omówię następujące zagadnienia:

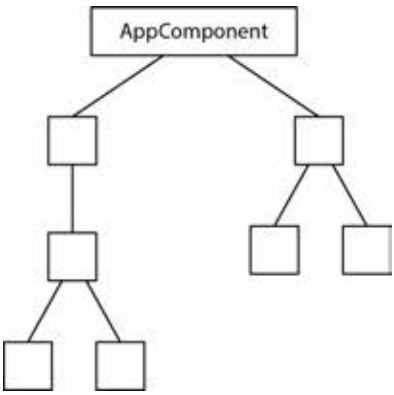
- interfejs CLI Angulara;
- bogaty ekosystem narzędzi dla Angulara w edytorze VS Code;
- tworzenie pierwszej aplikacji w Angularze;
- interakcje z Angularem;
- automatyzacja poleceń interfejsu CLI Angulara przy użyciu rozszerzenia **Nx Console.**

Podstawowe koncepcje i kontekst

Angular jest wieloplatformowym frameworkiem języka JavaScript działającym w różnych środowiskach, między innymi w sieci WWW, na serwerach, urządzeniach mobilnych i komputerach. Stanowi kolekcję bibliotek JavaScriptu pozwalających budować bardzo wydajne i skalowalne aplikacje internetowe. Architektura aplikacji utworzonych w Angularze jest oparta na hierarchii komponentów, które są dla niej podstawowymi elementami konstrukcyjnymi. Reprezentują one określony fragment strony WWW zwany **widokiem** (ang. *view*) i sterują jego działaniem. Przykładami komponentów mogą być:

- lista wpisów na blogu,
- formularz zgłaszania problemów,
- widżet wyświetlający pogodę.

Komponenty aplikacji napisanych w Angularze są zorganizowane logicznie w postaci drzewa (rysunek 1.1).



Rysunek 1.1. Drzewo komponentów

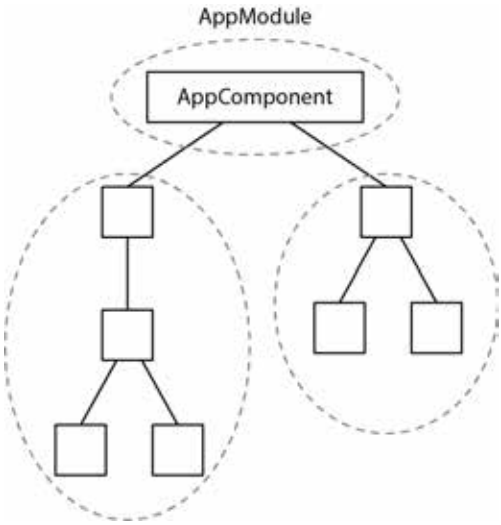
Zgodnie z konwencją aplikacja utworzona w Angularze ma zwykle jeden komponent główny o nazwie `AppComponent`. Każdy z komponentów drzewa może się komunikować i wchodzić w interakcje z innymi przy użyciu zdefiniowanego w każdym z nich interfejsu programowania aplikacji.

Aplikacja może mieć wiele funkcji, które są nazywane **modułami**. Każdy moduł stanowi blok oddzielnych funkcji odpowiadających konkretnej dziedzinie aplikacji lub następującemu w niej przepływowi pracy. Moduły Angulara pozwalają grupować komponenty o podobnych funkcjach (rysunek 1.2).

Na rysunku 1.2 okręgi narysowane linią przerywaną reprezentują moduły Angulara. Aplikacja ma zwyczajowo jednej moduł główny o nazwie `AppModule`. Poszczególne moduły aplikacji mogą importować inne moduły, by skorzystać z ich funkcji.

Zestaw funkcji modułu można dalej analizować pod kątem logiki prezentacyjnej i biznesowej. Komponenty Angulara powinny obsługiwać tylko logikę prezentacyjną, a funkcje biznesowe należy oddelegować do usług. Framework zapewnia komponentom dostęp usług przy użyciu wbudowanego mechanizmu **wstrzykiwania zależności** (ang. *dependency injection*, DI).

We frameworku wstrzykiwania zależności Angulara stosowane są obiekty specjalnego przeznaczania, zwane **iniektorami** (ang. *injectors*), ukrywające dużą część złożoności związanej z dostarczaniem zależności aplikacji. Programiści tworzący komponenty nie muszą znać szczegółów implementacji usług Angulara. Wystarczy, że zażądają usługi od iniektora.



Rysunek 1.2. Hierarchia modułów

Usługa Angulara musi być zgodna z **zasadą jednej odpowiedzialności** (ang. *single responsibility principle*) i nie powinna naruszać granic pomiędzy różnymi modułami. Oto kilka przykładów usług:

- dostęp do danych z API zaplecza przy użyciu protokołu HTTP,
- interakcja z pamięcią lokalną przeglądarki,
- rejestrowanie błędów,
- przekształcanie danych.

Programista tworzący aplikacje w Angularze nie musi uczyć się na pamięć sposobu tworzenia komponentów, modułów i usług. Na szczęście ma do pomocy interfejs CLI Angulara, który pozwala wykonać te zadania w wierszu poleceń.

Interfejs CLI Angulara

Angular CLI to narzędzie utworzone przez zespół rozwijający ten framework w celu usprawnienia pracy programistów budujących aplikacje w Angularze. Ukrywa ono skomplikowane kwestie związane z tworzeniem rusztowań i konfigurowaniem aplikacji, co pozwala programistom skupić się na tym, co umieją robić najlepiej — na kodowaniu! Zanim zaczniemy korzystać z CLI Angulara, musimy skonfigurować w swoim systemie wstępnie wymagane składniki:

- **Node.js** — środowisko uruchomieniowe języka JavaScript zbudowane na silniku v8 przeglądarki Chrome. Wersję o przedłużonym wsparciu (ang. *Long-Term Support*, LTS) można pobrać ze strony *https://nodejs.org*.
- **npm** — menedżer pakietów środowiska uruchomieniowego Node.js.

Narzędzie Angular CLI możemy wtedy zainstalować z wiersza poleceń przy użyciu programu `npm`:

```
npm install -g @angular/cli@v16-lts
```

Za pomocą opcji `-g` zainstalowaliśmy CLI globalnie, gdyż chcemy mieć możliwość tworzenia aplikacji na dowolnej ścieżce systemu operacyjnego.

Uwaga

W niektórych systemach operacyjnych instalowanie Angular CLI może wymagać uprawnień administratora.

Aby zweryfikować, czy interfejs CLI Angulara został poprawnie zainstalowany, możemy w wierszu poleceń wpisać:

```
ng version
```

Powyższe polecenie raportuje wersję zainstalowanego w systemie narzędzia Angular CLI. Jest ono dostępne z wiersza poleceń za pośrednictwem komendy `ng`, czyli binarnego pliku wykonywalnego interfejsu CLI Angulara. Przyjmuje on różne opcje, między innymi:

- `serve` — buduje i udostępnia aplikację.
- `build` — buduje aplikację.
- `test` — uruchamia testy jednostkowe aplikacji.
- `generate` — generuje nowy artefakt Angulara, na przykład komponent lub moduł.
- `add` — instaluje bibliotekę innego podmiotu zgodną z frameworkiem Angular.
- `new` — tworzy nową aplikację.

Wyżej wymienione opcje są najczęściej spotykane. Jeśli chcesz przejrzeć wszystkie dostępne komendy, wpisz w wierszu poleceń:

```
ng help
```

Powyższe polecenie wyświetli listę wszystkich komend dostępnych w interfejsie CLI Angulara.

Ekosystem narzędzi Angulara pełen jest rozszerzeń i programów użytkowych wspomagających tworzenie aplikacji. W następnym podrozdziale poznasz kilka z nich, które działają w edytorze VS Code.

Ekosystem narzędzi Angulara w edytorze VS Code

W witrynie **VS Code Marketplace** dostępnych jest wiele narzędzi rozszerzających ekosystem Angulara. W tym podrozdziale omówię kilka najpopularniejszych, bardzo wspomagających programowanie w tym frameworku:

- Nx Console,
- Angular Language Service,
- Angular Snippets,
- Angular Evergreen,
- Material Icon Theme.

Powyższa lista nie jest wyczerpująca; niektóre z tych rozszerzeń są zawarte w pakiecie *Angular Essentials*. Inne rozszerzenia Angulara dla VS Code można przejrzeć pod adresem *https://marketplace.visualstudio.com/search?term=angular&target=VSCode*.

Nx Console

Nx Console to rozszerzenie VS Code opracowane przez zespół Nrwl. Zapewnia ono graficzny interfejs użytkownika dla projektów tworzonych w Angularze i stanowi alternatywę dla większości poleceń interfejsu CLI Angulara. O rozszerzeniu tym dowiesz się więcej z podrozdziału „Automatyzacja poleceń interfejsu CLI Angulara przy użyciu Nx Console".

Angular Language Service

Rozszerzenie **Angular Language Service** ulepsza na kilka sposobów proces edytowania szablonów HTML aplikacji tworzonych w Angularze, między innymi dzięki:

- autouzupełnianiu kodu;
- pokazywaniu komunikatów o błędach kompilacji;
- implementacji technik przechodzenia do definicji.

Autouzupełnianie kodu to funkcja pozwalająca znaleźć odpowiednią właściwość lub metodę podczas pisania na klawiaturze. Działa na zasadzie wyświetlania listy podpowiedzi podczas wpisywania treści HTML (rysunek 1.3).

Rysunek 1.3. Uzupełnianie kodu

Na rysunku 1.3 widać, że gdy zaczęliśmy wpisywać znaki `descr`, rozszerzenie Angular Language Service zasugerowało użycie właściwości `description` komponentu. Warto wiedzieć, że uzupełnianie kodu działa tylko w przypadku publicznych właściwości i metod komponentu.

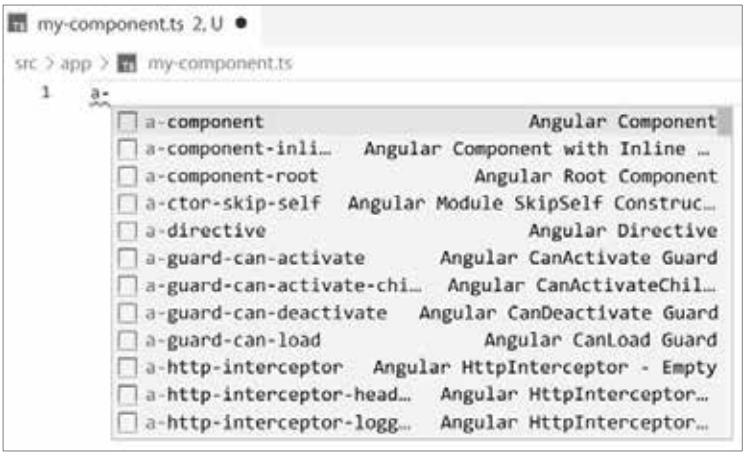
Jednym z najczęstszych problemów podczas wytwarzania aplikacji internetowej jest wykrycie w niej błędów, zanim trafi na produkcję. Problem ten można częściowo rozwiązać dzięki kompilatorowi Angulara, który jest uruchamiany podczas budowania aplikacji dla środowiska produkcyjnego. Angular Language Service idzie jednak dalej i wyświetla komunikaty o błędach kompilacji dużo wcześniej, nim aplikacja zostanie poddana kompilacji (rysunek 1.4).

Rysunek 1.4. Komunikat to błędzie kompilacji

Jeśli na przykład przypadkowo błędnie wpiszemy nazwę właściwości lub metody komponentu, Angular Language Service wyświetli odpowiedni komunikat o błędzie.

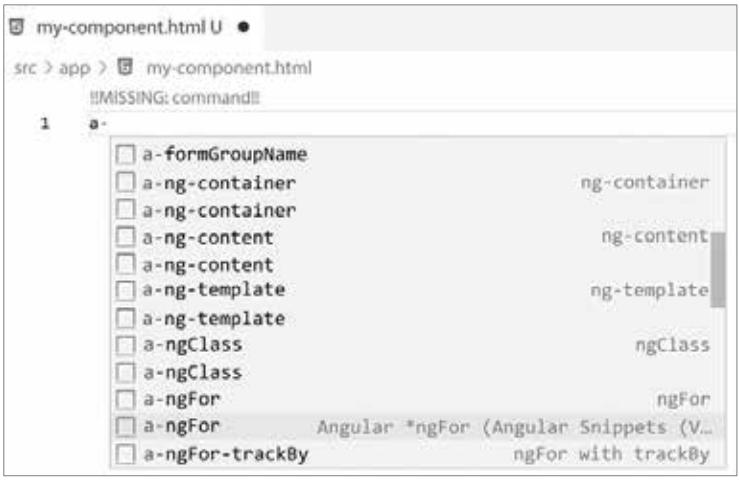
Angular Snippets

Rozszerzenie **Angular Snippets** zawiera kolekcję przykładów kodu napisanego za pomocą Angulara w językach TypeScript i HTML. Umożliwia ono tworzenie komponentów, modułów i usług w pustym pliku TypeScriptu (rysunek 1.5).



Rysunek 1.5. Fragment kodu tworzącego nowy komponent Angulara

W szablonach HTML rozszerzenie to pozwala tworzyć przydatne artefakty Angulara, na przykład dyrektywę `*ngFor` do przechodzenia w pętli po liście HTML (rysunek 1.6).



Rysunek 1.6. Fragment kodu z dyrektywą *ngFor

Ze względu na dużą popularność i możliwości interfejsu CLI Angulara wydaje się on wygodniejszym narzędziem do generowania artefaktów w języku TypeScript. Angular Snippets sprawdza się jednak doskonale w przypadku szablonów HTML, w których jest więcej rzeczy do zapamiętania.

Material Icon Theme

Ostatnie rozszerzenie z naszej listy w niewielkim stopniu przyczynia się do zwiększenia wydajności programisty. Ułatwia za to odnajdowanie funkcji edytora VS Code i poprawia wrażenia estetyczne dzięki zmianie motywu ikon.

Material Icon Theme zawiera mnóstwo ikonek zgodnych ze stylem **Material Design** firmy Google. Każdy typ pliku w projekcie jest automatycznie rozpoznawany i wyświetlana jest przy nim odpowiednia ikonka. Moduły Angulara są na przykład oznaczane czerwonym logo Angulara, komponenty zaś niebieskim.

VS Code ma domyślny motyw ikonek plików o nazwie **Seti**. Po zainstalowaniu rozszerzenia Material Icon Theme pojawi się monit z prośbą o wybranie motywu, który ma być aktywny (rysunek 1.7).

Rysunek 1.7. Wybór motywu ikonek plików

Po wybraniu motywu Material Icon Theme ikonki w bieżącym projekcie Angulara zostaną automatycznie zaktualizowane.

Uwaga

Motyw Material Icon Theme jest instalowany i stosowany globalnie w programie VS Code, więc nie trzeba go aktywować oddzielnie w każdym projekcie opartym na CLI Angulara.

Od tej pory po otworzeniu projektu Angulara od razu rozpoznasz typ każdego pliku, nawet jeśli jego nazwa nie mieści się na ekranie.

Omówienie projektu

W tym projekcie utworzymy od podstaw aplikację w Angularze przy użyciu interfejsu CLI. Następnie wejdziemy w interakcje z podstawowymi funkcjami frameworka Angular i dokonamy w aplikacji drobnej zmiany. Na koniec nauczysz się budować i udostępniać aplikację przy użyciu rozszerzenia Nx Console.

Czas budowania: 15 minut.

Rozpoczęcie pracy

Do wykonania tego projektu potrzebne będą następujące programy:

- **Git** — darmowy rozproszony system kontroli wersji o otwartym kodzie źródłowym. Można go pobrać ze strony *https://git-scm.com*.
- **VS Code** — edytor kodu źródłowego dostępny pod adresem *https://code.visualstudio.com*.
- **Angular CLI** — interfejs wiersza poleceń Angulara przedstawiłem w podrozdziale „Podstawowe koncepcje i kontekst”.
- **Pliki przykładów** — przykłady kodu z tego rozdziału można znaleźć w katalogu *r01* w archiwum dostępnym pod adresem *https://ftp.helion.pl/przyklady/angdz3.zip*.

Tworzenie pierwszej aplikacji w Angularze

Aby utworzyć w Angularze całkiem nową aplikację, musimy wydać polecenie interfejsu CLI `ng new` i podać w opcji nazwę aplikacji:

```
ng new my-app
```

Polecenie `ng new` służy do tworzenia nowej aplikacji lub obszaru roboczego Angulara. Obszar roboczy to projekt Angular CLI zawierający jedną lub kilka aplikacji, przy czym niektóre z nich mogą być bibliotekami. Za pomocą polecenia `ng new` utworzymy więc domyślnie obszar roboczy Angulara z jedną aplikacją.

W powyższym poleceniu nazwa aplikacji Angulara to `my-app`. Po jego wydaniu CLI Angulara zada nam kilka pytań, aby zebrać jak najwięcej informacji o tym, jakiego rodzaju aplikację chcemy utworzyć:

1. Na początku pojawi się pytanie o to, czy chcemy włączyć analizy Angulara:

   ```
   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. (y/N)
   ```

   CLI Angulara zadaje to pytanie tylko raz, podczas tworzenia pierwszego projektu, i stosuje to ustawienie w całym systemie. Możemy je jednak zmienić w poszczególnych obszarach roboczych Angulara.

2. Następnie pojawi się pytanie o to, czy chcemy włączyć w aplikacji routing:

   ```
   Would you like to add Angular routing? (y/N)
   ```

   Chodzi tu o nawigowanie pomiędzy komponentami aplikacji przy użyciu adresu URL. Routing w tym projekcie nie jest nam potrzebny, naciśnij więc klawisz *Enter*, aby zaakceptować domyślną wartość tego ustawienia.

3. Potem CLI Angulara poprosi o wybranie formatu arkusza stylów, z którego będziemy korzystać w aplikacji:

   ```
   Which stylesheet format would you like to use? (Use arrow keys)
   ```

   Wybierz jeden format arkuszy stylów z listy i naciśnij klawisz *Enter*.

CLI Angulara zainicjuje proces tworzenia aplikacji, składający się z następujących etapów:

- generowanie niezbędnej struktury typowego projektu Angular CLI;
- instalowanie wymaganych zależności npm i pakietów Angulara;
- inicjowanie repozytorium Git w projekcie Angular CLI.

Proces ten zależnie od szybkości sieci może zająć trochę czasu. Po jego zakończeniu w katalogu, w którym zostało wydane polecenie `ng new` interfejsu CLI Angulara, pojawi się nowy folder, o nazwie *my-app*.

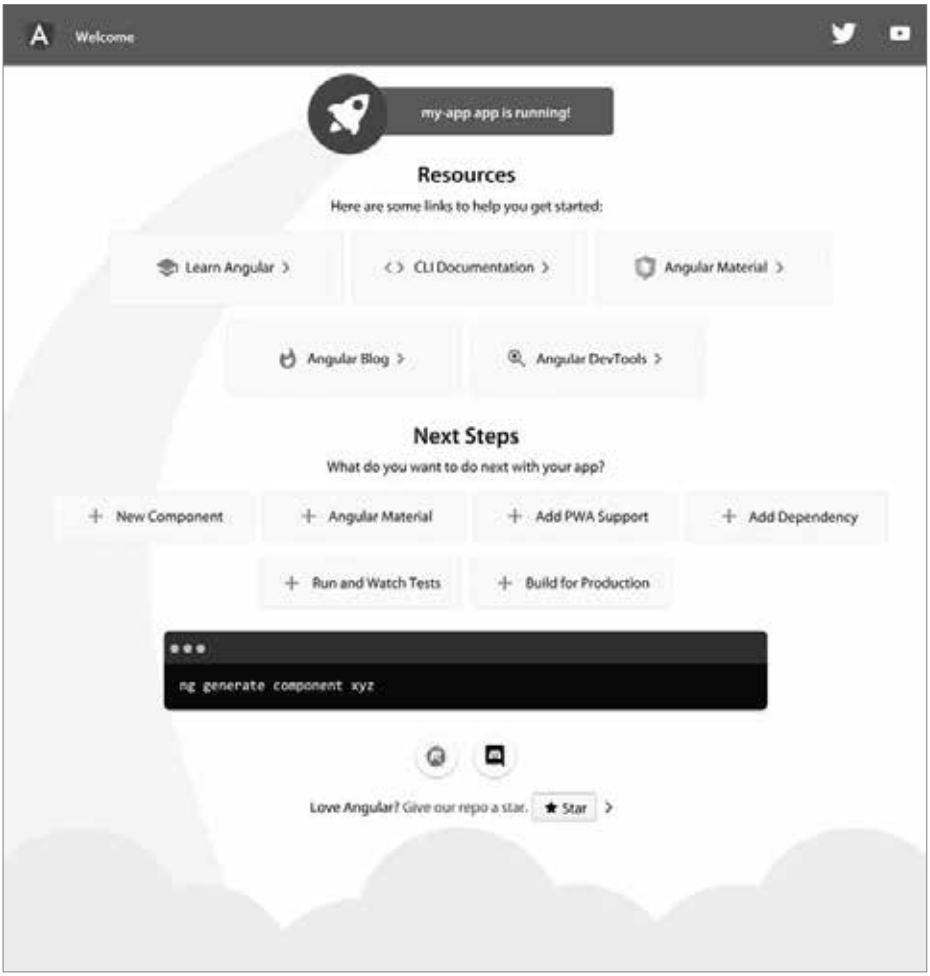
Czas uruchomić w końcu aplikację i zobaczyć, jak działa:

1. Otwórz okno terminala i przejdź do folderu *my-app*.
2. Wydaj następujące polecenie interfejsu CLI Angulara:

   ```
   ng serve
   ```

   Powyższe polecenie spowoduje zbudowanie aplikacji i uruchomienie wbudowanego serwera WWW, co pozwoli ją podejrzeć. Serwer WWW jest uruchamiany w trybie monitorowania (ang. *watch mode*), aplikacja będzie

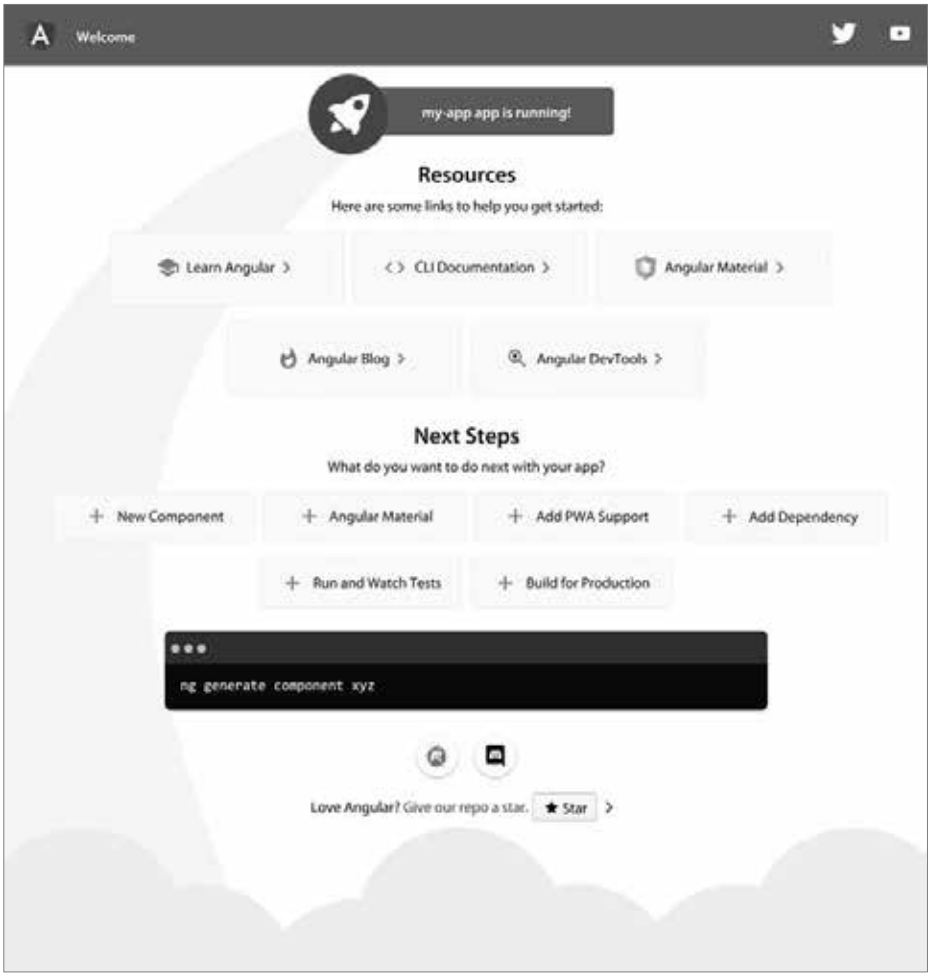
więc automatycznie budowana ponownie za każdym razem, gdy zmienimy jej kod. Za pierwszym razem zbudowanie aplikacji zajmuje sporo czasu, więc musimy uzbroić się w cierpliwość. O tym, że proces zakończył się bezbłędnie, dowiemy się, gdy zobaczymy w oknie terminala komunikat z rysunku 1.8.



Rysunek 1.8. Dane wyjściowe procesu budowania aplikacji w Angularze

3. Uruchom swoją ulubioną przeglądarkę i przejdź pod adres *http://localhost:4200*, aby zobaczyć całkiem nową aplikację zbudowaną w Angularze (rysunek 1.9).

CLI Angulara tworzy domyślnie ograniczoną do minimum aplikację, by zapewnić punkt startowy dla naszego projektu. Zawiera ona kilka gotowych styli CSS i treść HTML, którą nauczymy się zmieniać zgodnie z naszymi założeniami w kolejnym podrozdziale.



Rysunek 1.9. Minimalistyczna aplikacja utworzona w Angularze

Interakcje z Angularem

Podczas pracy z Angularem prawdziwa zabawa zaczyna się wtedy, gdy weźmiemy się do pracy z samym frameworkiem. W końcu chodzi o to, by zrozumieć, jak działa Angular, i zacząć pisać kod aplikacji.

Kod źródłowy aplikacji mieści się w folderze *src\app* znajdującym się w głównym katalogu projektu utworzonego przy użyciu interfejsu CLI Angulara. Folder ten zawiera wszystkie pliki potrzebne do zbudowania i przetestowania naszej aplikacji, łącznie z komponentem i modułem. Ten pierwszy jest komponentem głównym aplikacji:

app.component.ts

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'my-app';
}
```

Komponent Angulara ma następujące właściwości:

- `selector` — unikatowa nazwa pozwalająca zidentyfikować i zadeklarować komponent w treści HTML. Jest to znacznik języka HTML, przypominający jego elementy natywne, na przykład `<app-root></app-root>`.

Wskazówka

CLI Angulara domyślnie dodaje do selektorów komponentów prefiks `app-`. Podczas tworzenia od początku nowej aplikacji w interfejsie CLI Angulara w opcji `--prefix` możemy określić własny przedrostek. Nadaje się go na przykład na podstawie nazwy firmy lub konkretnego produktu. Pozwala on uniknąć kolizji nazw z innymi bibliotekami i modułami.

- `templateUrl` — ścieżka do pliku z treścią HTML zwaną szablonem komponentu.
- `styleUrls` — lista ścieżek do plików arkuszy stylów CSS komponentu.

Powyższe właściwości definiuje się przy użyciu dekoratora `@Component`. Jest to funkcja, która dekoruje klasę TypeScriptu komponentu oraz identyfikuje ją jako komponent Angulara. Właściwość `title` klasy `AppComponent` jest publiczna, zawiera ciąg tekstowy i może być użyta w szablonie komponentu.

Właściwości głównego modułu aplikacji definiuje się przy użyciu podobnego dekoratora o nazwie `@NgModule`:

app.module.ts

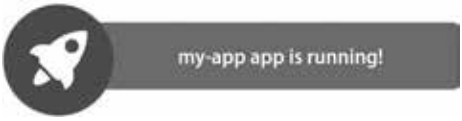
```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppComponent } from './app.component';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule
```

```
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

W dekoratorze modułu Angulara określa się zbiór właściwości służących do konfigurowania tego modułu. Najczęściej spotykane są następujące właściwości:

- `declarations` — definiuje komponenty Angulara będące częścią modułu. Do tablicy `declarations` *musi* być dodany każdy istniejący w module Angulara komponent.
- `imports` — definiuje inne moduły Angulara, które zawierają potrzebne w bieżącym module funkcje.

Czas postawić pierwszy krok i zmodyfikować kod naszej aplikacji budowanej w Angularze. Zmienimy komunikat powitalny wyświetlany podczas startu aplikacji (rysunek 1.10) na coś bardziej zrozumiałego.



Rysunek 1.10. Komunikat powitalny

Przede wszystkim musimy znaleźć miejsce, w którym jest zadeklarowany komunikat pokazany na rysunku 1.10. Komponentem ładowanym domyślnie podczas startu aplikacji jest komponent główny.

Wskazówka

Właściwość `bootstrap` modułu głównego aplikacji wskazuje komponent wyświetlany podczas uruchamiania aplikacji. Rzadko kiedy potrzeba ją zmieniać. Selektor tego komponentu jest używany domyślnie w pliku *index.html*.

Komunikat powinien być zadeklarowany w pliku *app.component.ts*. Zajrzyjmy tam:

1. Otwórz edytor VS Code i wybierz z głównego menu polecenie *Plik/Otwórz folder...*.
2. Znajdź folder *my-app* utworzonej przez nas aplikacji i wybierz go.
3. W panelu *Eksplorator* przejdź do folderu *src\app* i wybierz plik *app.component.ts*.
4. Znajdź właściwość `title` klasy `AppComponent` i zmień wartość tej właściwości na `Projekty w Angularze`:

   ```
   title = 'Projekty w Angularze';
   ```

5. Jeśli aplikacja nie działa, w oknie terminala wydaj plecenie `ng serve` i przejdź w przeglądarce pod adres *http://localhost:4200*. Nasza aplikacja powinna teraz wyświetlać komunikat powitalny taki jak na rysunku 1.11.

Rysunek 1.11. Zmodyfikowany komunikat powitalny

Właściwość `title` jest powiązana z szablonem komponentu głównego. Jeśli otworzymy plik *app.component.html* i przejdziemy do wiersza 344., zobaczymy następujący kod HTML:

```
<span>{{ title }} app is running!</span>
```

Znaki {{}}, w które jest ujęta właściwość `title`, to składnia **interpolacji**. W jej trakcie Angular odczytuje wartość ujętej w nawiasy właściwości komponentu, konwertuje ją na tekst i wyświetla go na ekranie.

Zmieńmy treść tego wiersza następująco:

```
<span>Aplikacja {{ title }} działa!</span>
```

Zmieni się wówczas również komunikat powitalny (rysunek 1.12).

Rysunek 1.12. Spolonizowana wersja komunikatu powitalnego

CLI Angulara ma bogatą kolekcję poleceń pomagających w codziennych pracach projektowych. Wielu programistom korzystanie z wiersza poleceń sprawia jednak trudność i preferują podejście bardziej wizualne. Z następnego rozdziału dowiesz się, jak korzystać z rozszerzenia Nx Console, dostarczającego graficzny interfejs użytkownika dla narzędzia Angular CLI.

Automatyzacja poleceń interfejsu CLI Angulara przy użyciu rozszerzenia Nx Console

Angular CLI to narzędzie wiersza poleceń. Ma ono wiele komend, z których każda może mieć wiele opcji i parametrów w zależności od zadania, które chcemy wykonać. Nauczenie się tych poleceń i ich opcji na pamięć może być zniechęcające i czasochłonne.

W tej sytuacji warto skorzystać z ekosystemu narzędzi Angulara. VS Code Marketplace zawiera wiele przydatnych rozszerzeń, które możemy zainstalować po to, by nam pomagały podczas programowania z użyciem Angulara. Jednym z takich rozszerzeń jest Nx Console. Zapewnia ono interfejs użytkownika alternatywny wobec Angular CLI. Aby zainstalować to rozszerzenie w swoim środowisku, wykonaj następujące kroki:

1. Otwórz VS Code i kliknij na pasku bocznym przycisk *Rozszerzenia* (rysunek 1.13).

Rysunek 1.13. Rozszerzenia edytora VS Code

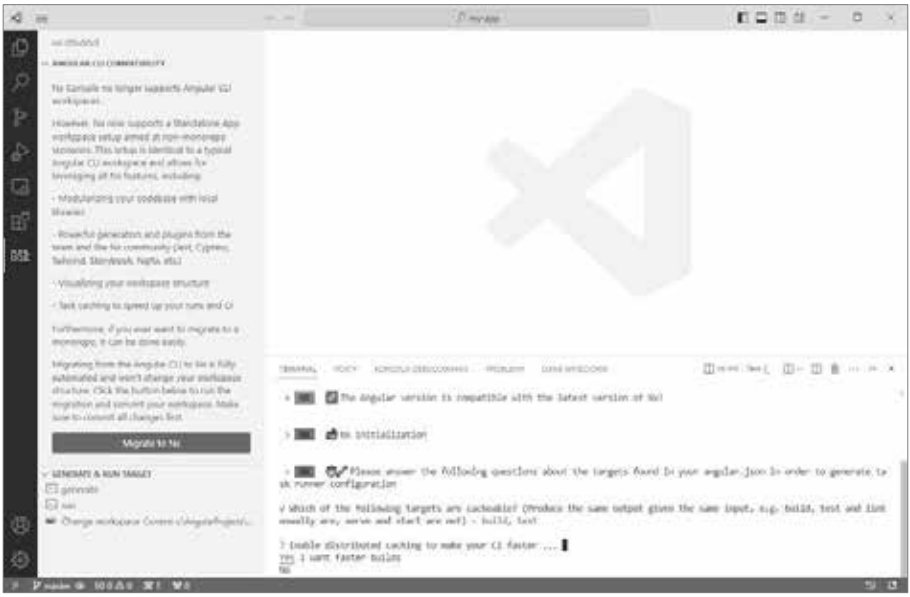
2. W panelu *Rozszerzenia*, który się pojawi, wpisz **`Nx Console`**.
3. Aby zainstalować rozszerzenie Nx Console, kliknij przy pierwszej pozycji przycisk *Zainstaluj*.

Rozszerzenie Nx Console instaluje się w środowisku globalnie, więc będziemy mogli go używać w każdym projekcie Angulara. Reprezentuje ono w sposób graficzny najczęściej używane polecenia własnego interfejsu CLI. Obsługuje między innymi następujące polecenia (w nawiasach podaję odpowiadające im komendy wiersza poleceń):

- *generate* — generuje nowe artefakty Angulara, między innymi komponenty i moduły (`npx nx g`).
- *run* — uruchamia cel architektoniczny zdefiniowany w pliku konfiguracyjnym *project.json* obszaru roboczego Nx Console (`npx nx run`).
- *build* — buduje aplikację (`npx nx build`).
- *serve* — buduje i udostępnia aplikację (`npx nx serve`).
- *test* — uruchamia testy jednostkowe aplikacji (`npx nx test`).

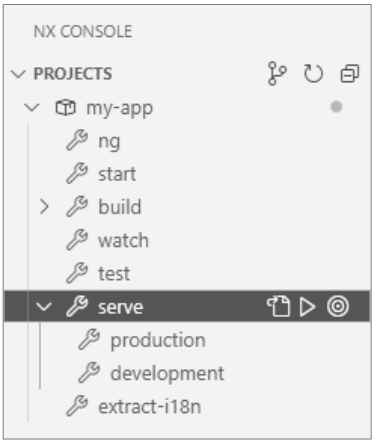
Za pomocą rozszerzenia Nx Console można osiągnąć prawie wszystko to, co można zrobić w interfejsie CLI Angulara. Prawdziwą korzyścią jest to, że programista nie musi pamiętać wszystkich opcji narzędzia wiersza poleceń, gdyż są one dostępne w interfejsie graficznym. Zobaczmy, jak się nim posługiwać:

1. W edytorze VS Code otwórz folder *my-app* i kliknij przycisk *Nx Console* znajdujący się na pasku bocznym. W panelu Nx Console pojawi się komunikat o braku zgodności z projektami utworzonymi przy użyciu CLI Angulara. Należy klinąć przycisk *Migrate to Nx* (migruj do Nx), a w terminalu VS Code odpowiedzieć na kilka pytań (rysunek 1.14).



Rysunek 1.14. Panel Nx Console

2. Z panelu *Projects* (projekty) wybierz polecenie *serve* (obsłuż) i kliknij przycisk odtwarzania, aby je uruchomić (rysunek 1.15).



Rysunek 1.15. Polecenie serve

3. W programie VS Code pod oknem edytora otworzy się zintegrowany terminal, w którym zostanie wykonane polecenie `npx nx run my-app:serve` (rysunek 1.16).

Rysunek 1.16. Terminal zintegrowany z VS Code

Podobne polecenie uruchamiamy przy użyciu interfejsu CLI Angulara w oknie terminala.

Nx Console uruchamia wewnętrznie polecenia interfejsu CLI Angulara przy użyciu **zadań** (ang. *tasks*). Są one wbudowanym w VS Code mechanizmem pozwalającym uruchamiać skrypty lub procesy zewnętrzne bez bezpośredniej interakcji z wierszem poleceń.

Rozszerzenie Nx Console sprawdza się świetnie, gdyż zwalnia użytkownika z konieczności uczenia się na pamięć poleceń interfejsu CLI Angulara. Witryna VS Code Marketplace zawiera dużo więcej rozszerzeń uzupełniających działanie narzędzia Nx Console.

Podsumowanie

W tym rozdziale omówiłem podstawowe zasady obowiązujące we frameworku Angular i pokrótce objaśniłem jego architekturę. Przedstawiłem kilka popularnych rozszerzeń VS Code, ułatwiających programistom pracę z Angularem.

Potem pokazałem, jak korzystać z interfejsu CLI Angulara, który jest wszechstronnym narzędziem w jego ekosystemie. Za pomocą tego interfejsu utworzyliśmy rusztowanie nowej aplikacji i zbudowaliśmy ją od podstaw. Mamy też za sobą pierwszą interakcję z kodem w Angularze, która polegała na zmodyfikowaniu komponentu typowej aplikacji utworzonej przy użyciu interfejsu CLI. Na koniec zainstalowaliśmy narzędzie Nx Console i pokazałem, jak zbudować w nim aplikację.

W następnym rozdziale przyjrzymy się usłudze Angular Router i dowiesz się, jak za jej pomocą i przy użyciu generatora witryn statycznych o nazwie Scully utworzyć osobisty blog.

Pytania sprawdzające

Odpowiedz na pytania sprawdzające:

1. Jakie są podstawowe elementy konstrukcyjne aplikacji tworzonych z użyciem Angulara?
2. W jaki sposób grupujemy komponenty o podobnych funkcjach?
3. Co obsługuje zadania logiki biznesowej w aplikacjach budowanych w Angularze?
4. Jakim poleceniem interfejsu CLI Angulara tworzymy nową aplikację?
5. Jakim poleceniem interfejsu CLI Angulara udostępniamy aplikację?
6. W jaki sposób deklarujemy komponent Angulara w kodzie HTML?
7. W jaki sposób deklarujemy komponenty Angulara w module?
8. Za pomocą jakiej składni wiążemy dane tekstowe w szablonach HTML?
9. Jaką korzyść przynosi używanie rozszerzenia Nx Console?
10. Za pomocą jakiego rozszerzenia dokonamy analizy statycznej kodu napisanego w Angularze?

Materiały dodatkowe

Oto kilka łączy, dzięki którym możesz pogłębić zdobytą w tym rozdziale wiedzę:

- Wprowadzenie do podstawowych koncepcji Angulara: *https://angular.io/guide/architecture*
- Interpolacja: *https://angular.io/guide/interpolation*
- Nx Console: *https://nx.dev/core-features/integrate-with-editors#vscode-plugin:-nx-console*
- Angular Essentials: *https://marketplace.visualstudio.com/items?itemName=johnpapa.angular-essentials*
- Angular Evergreen: *https://expertlysimple.io/get-evergreen*

Angular: Twoja droga do tworzenia nowoczesnych aplikacji!

Angular jest ulubionym narzędziem programistów, którzy cenią wydajność pracy i chcą z niej czerpać pełną satysfakcję. Umożliwia tworzenie aplikacji działających na wielu platformach, w sieci WWW, na komputerach i urządzeniach mobilnych. Co więcej, pozwala na użycie języka TypeScript, który lepiej niż JavaScript nadaje się do programowania profesjonalnych aplikacji internetowych. Angular zapewnia również możliwość korzystania z wielu nowoczesnych bibliotek.

Dzięki temu praktycznemu przewodnikowi sprawdzisz, jak działa Angular podczas tworzenia dziesięciu zróżnicowanych funkcjonalnych aplikacji internetowych. Nauczysz się też integrować go z różnymi bibliotekami i narzędziami, takimi jak Angular Router, Scully, Electron, wątki robocze usług Angulara czy narzędzia Nx do zarządzania repozytoriami monolitycznymi. Poszczególnych technologii użyjesz do tworzenia ciekawych projektów: aplikacji pogodowej w technice PWA, mobilnej aplikacji do geotagowania zdjęć, biblioteki komponentów interfejsu użytkownika i innych. Dowiesz się też, jak dostosować interfejs CLI Angulara do swoich potrzeb.

W książce między innymi:

- konfiguracja aplikacji Angulara
- tworzenie jednostronicowych aplikacji za pomocą Jamstack i Scully
- budowa systemu śledzenia problemów przy użyciu Typed Reactive Forms
- tworzenie aplikacji w repozytorium monolitycznym z zastosowaniem narzędzi Nx
- programowanie aplikacji mobilnych z wykorzystaniem frameworka Ionic
- własne schematy rozszerzające interfejs wiersza poleceń Angulara

Aristeidis Bampakos jest greckim programistą z wieloletnim doświadczeniem. Kieruje zespołem tworzącym strony WWW w Angularze. W 2020 roku otrzymał tytuł Google Developer Expert (GDE) dla tej platformy. Jego pasją jest pomaganie programistom w rozwoju, obecnie uczestniczy w tłumaczeniu oficjalnej dokumentacji Angulara na język grecki.

Helion
helion.pl
HELION SA
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl
KOD KORZYŚCI
Sięgnij po więcej!
ISBN 978-83-289-0809-3
9 788328 908093
Cena: 69,00 zł
<packt>