

# Spis treści

Przedmowa 11

Podziękowania 13

O książce 15

O autorach 19

Rozdział 1. Wprowadzenie do frameworku Angular 21

1.1. Dlaczego do tworzenia aplikacji internetowych wybierać Angular? 22

1.2. Dlaczego tworzyć aplikacje w TypeScriptie, a nie w JavaScriptcie 23

1.3. Przegląd frameworku Angular 24

1.4. Wprowadzenie do CLI Angulara 28

1.4.1. Generowanie nowego projektu Angulara 29

1.4.2. Kompilacje programistyczne i produkcyjne 31

1.5. Porównanie kompilacji JIT i AOT 33

1.5.1. Tworzenie paczek za pomocą opcji -prod 34

1.5.2. Generowanie paczek na dysku 35

1.6. Wprowadzenie do przykładowej aplikacji ngAuction 35

Podsumowanie 38

Rozdział 2. Główne artefakty aplikacji napisanych w Angularze 39

2.1. Komponenty 39

2.2. Usługi 42

2.3. Dyrektywy 43

2.4. Potoki 45

2.5. Moduły 46

2.5.1. Moduły funkcyjne 46

2.6. Wiązanie danych 50

2.6.1. Wiązanie właściwości i zdarzeń	50
2.6.2. Jedno- i dwukierunkowe wiązanie danych w działaniu	51
2.7. Część praktyczna: rozpoczynamy tworzenie aplikacji ngAuction	55
2.7.1. Wstępna konfiguracja projektu dla aplikacji ngAuction	55
2.7.2. Generowanie komponentów dla aplikacji ngAuction	57
2.7.3. Komponent aplikacji	57
2.7.4. Komponent paska nawigacyjnego	59
2.7.5. Komponent wyszukiwania	60
2.7.6. Komponent stopki	61
2.7.7. Komponent karuzeli	61
2.7.8. Komponent strony głównej	64
Podsumowanie	65
Rozdział 3. Podstawy routera Angulara	67
3.1. Podstawy routingu	68
3.2. Strategie lokalizacji	69
3.2.1. Nawigacja oparta na znaku kratki	70
3.2.2. Nawigacja oparta na interfejsie History API	70
3.3. Bloki konstrukcyjne nawigacji po stronie klienta	71
3.4. Nawigacja do tras za pomocą metody navigate()	76
3.5. Przekazywanie danych do tras	77
3.5.1. Wyodrębnianie parametrów z ActivatedRoute	78
3.5.2. Przekazywanie do trasy parametrów zapytania	80
3.6. Trasy podrzędne	81
3.7. Część praktyczna: dodanie nawigacji do aplikacji aukcji internetowych	86
3.7.1. Usługa ProductService	87
3.7.2. Komponent ProductItemComponent	88
3.7.3. Komponent HomeComponent	89

3.7.4. Komponent StarsComponent	91
3.7.5. Komponent ProductDetailComponent	94
Podsumowanie	96
Rozdział 4. Zaawansowana konfiguracja routera	97
4.1. Strzeżenie tras	98
4.1.1. Implementowanie strażnika CanActivate	99
4.1.2. Implementowanie strażnika CanDeactivate	101
4.1.3. Implementowanie strażnika Resolve	103
4.2. Tworzenie aplikacji SPA z wieloma gniazdami routera	107
4.2.1. Moduły leniwie ładowane	109
4.2.2. Ładowarki wstępne	113
Podsumowanie	114
Rozdział 5. Wstrzykiwanie zależności we frameworku Angular	115
5.1. Wzorzec Wstrzykiwanie Zależności	116
5.2. Korzyści płynące ze wstrzykiwania zależności w aplikacjach napisanych w Angularze	117
5.2.1. Luźne powiązania i wielokrotne wykorzystywanie	117
5.2.2. Testowalność	119
5.3. Wstrzykiwacze i dostawcy	119
5.4. Prosta aplikacja ze wstrzykiwaniem zależności frameworku Angular	122
5.4.1. Wstrzyknięcie usługi produktowej	122
5.4.2. Wstrzyknięcie usługi HttpClient	125
5.5. Ułatwione przełączanie wstrzykiwaczy	127
5.6. Deklarowanie dostawców za pomocą właściwości useFactory i useValue	131
5.6.1. Korzystanie z klasy InjectionToken	133
5.6.2. Wstrzykiwanie zależności w aplikacji zmodularyzowanej	134
5.7. Dostawcy w modułach ładowanych leniwie	134
5.8. Dostawcy w modułach ładowanych gorliwie	136

5.9. Część praktyczna: użycie komponentów biblioteki Angular Material w aplikacji ngAuction 138

5.9.1. Krótki przegląd biblioteki Angular Material 139

5.9.2. Dodanie biblioteki AM do projektu 141

5.9.3. Dodanie modułu funkcyjnego z komponentami AM 142

5.9.4. Modyfikacja wyglądu komponentu NavbarComponent 143

5.9.5. Modyfikacja interfejsu użytkownika komponentu SearchComponent 146

5.9.6. Zastąpienie karuzeli obrazem 148

5.9.7. Kolejne poprawki odstępów 148

5.9.8. Użycie mat-card w komponencie ProductItemComponent 148

5.9.9. Dodanie stylów do komponentu HomeComponent 149

Podsumowanie 149

Rozdział 6. Programowanie reaktywne we frameworku Angular 151

6.1. Obsługa zdarzeń bez strumieni obserwowalnych 152

6.2. Przekształcanie zdarzeń DOM w strumienie obserwowalne 154

6.3. Obsługa zdarzeń obserwowalnych za pomocą Forms API 156

6.4. Odrzucanie wyników niechcianych żądań HTTP za pomocą operatora switchMap 158

6.5. Korzystanie z potoku AsyncPipe 161

6.6. Strumienie obserwowalne i router 165

Podsumowanie 168

Rozdział 7. Tworzenie układu stron za pomocą biblioteki Flex Layout 169

7.1. Biblioteka Flex Layout i usługa ObservableMedia 170

7.1.1. Korzystanie z dyrektyw biblioteki Flex Layout 171

7.1.2. Usługa ObservableMedia 175

7.2. Część praktyczna: przepisanie kodu aplikacji ngAuction 177

7.2.1. Po co przepisywać aplikację ngAuction od zera? 177

7.2.2. Generowanie nowej aplikacji ngAuction 180

7.2.3. Tworzenie niestandardowego motywu Angular Material za pomocą Sass 180

7.2.4. Dodawanie paska narzędzi do komponentu najwyższego poziomu	183
7.2.5. Tworzenie usługi produktowej	186
7.2.6. Tworzenie modułu strony głównej	188
7.2.7. Konfigurowanie tras	193
7.2.8. Uruchomienie aplikacji ngAuction	193
Podsumowanie	194
Rozdział 8. Implementowanie komunikacji komponentów	195
8.1. Komunikacja między komponentami	196
8.2. Właściwości wejściowe i wyjściowe	196
8.2.1. Właściwości wejściowe	197
8.2.2. Właściwości wyjściowe i zdarzenia niestandardowe	199
8.3. Implementowanie wzorca projektowego Mediator	203
8.3.1. Użycie jako mediatora wspólnego komponentu nadrzędnego	203
8.3.2. Użycie jako mediatora wstrzykiwalnej usługi	208
8.4. Udostępnianie interfejsu API komponentu potomnego	213
8.5. Rzutowanie szablonów w czasie działania aplikacji za pomocą dyrektywy ngContent	216
8.5.1. Tryby hermetyzacji widoków	218
8.5.2. Rzutowanie na wiele obszarów	220
Podsumowanie	222
Rozdział 9. Mechanizm wykrywania zmian i cykl życia komponentu	223
9.1. Ogólny przegląd działania mechanizmu wykrywania zmian	224
9.1.1. Strategie wykrywania zmian	225
9.1.2. Profilowanie wykrywania zmian	227
9.2. Cykl życia komponentów	227
9.2.1. Przechwytywanie zmian w zaczeple ngOnChanges	230
9.2.2. Przechwytywanie zmian w zaczeple ngDoCheck	233
9.3. Część praktyczna: dodanie widoku produktu do aplikacji ngAuction	236

9.3.1. Tworzenie komponentów i modułu produktu	236
9.3.2. Implementowanie komponentu produktu	238
9.3.3. Implementowanie komponentu szczegółów produktu	241
9.3.4. Implementowanie komponentu sugerowanych produktów	242
Podsumowanie	244
Rozdział 10. Wprowadzenie do interfejsu API formularzy	245
10.1. Dwa interfejsy API formularzy	246
10.2. Formularze oparte na szablonach	246
10.2.1. Dyrektywy formularzy	246
10.2.2. Zastosowanie opartego na szablonach API do formularzy HTML	248
10.3. Formularze reaktywne	251
10.3.1. Model formularza	252
10.3.2. Dyrektywy formularzy reaktywnych	254
10.3.3. Zastosowanie reaktywnego API do formularzy HTML	256
10.3.4. Dynamiczne dodawanie kontrolek do formularza	258
10.4. Podsumowanie dyrektyw Forms API	259
10.5. Aktualizacja na podstawie danych	261
10.6. Korzystanie z klasy FormBuilder	263
Podsumowanie	263
Rozdział 11. Walidacja formularzy	265
11.1. Korzystanie z wbudowanych walidatorów	266
11.2. Kontrolowanie momentu rozpoczęcia walidacji	269
11.3. Walidatory niestandardowe w formularzach reaktywnych	270
11.4. Walidacja grupy kontrolek	273
11.5. Sprawdzanie statusu i prawidłowości kontrolki formularza	276
11.5.1. Dotknięte i niedotknięte kontrolki formularza	276
11.5.2. Pola czyste i brudne	277

11.5.3. Pola oczekujące	277
11.6. Dynamiczne zmienianie walidatorów w formularzach reaktywnych	277
11.7. Walidatory asynchroniczne	279
11.8. Niestandardowe walidatory w formularzach opartych na szablonach	281
11.9. Część praktyczna: dodanie formularza wyszukiwania do aplikacji ngAuction	282
11.9.1. Komponent formularza wyszukiwania	283
11.9.2. Komponent wyników wyszukiwania	287
11.9.3. Refaktoryzacja pozostałego kodu	288
Podsumowanie	291
Rozdział 12. Interakcja z serwerami przy użyciu protokołu HTTP	293
12.1. Przegląd usługi HttpClient	294
12.2. Odczytywanie pliku JSON za pomocą usługi HttpClient	295
12.3. Tworzenie serwera WWW za pomocą frameworków Node i Express oraz języka TypeScript	298
12.3.1. Tworzenie prostego serwera WWW	299
12.3.2. Serwowanie danych w formacie JSON	301
12.4. Łączenie frameworku Angular i serwera Node	303
12.4.1. Zasoby statyczne na serwerze	303
12.4.2. Konsumowanie danych JSON w aplikacjach Angulara	305
12.4.3. Konfigurowanie serwera proxy klienta	308
12.4.4. Subskrybowanie strumieni obserwowalnych za pomocą potoku async	309
12.4.5. Wstrzykiwanie HttpClient do usługi	310
12.4.6. Wdrażanie aplikacji Angulara na serwerze za pomocą skryptów npm	312
12.5. Przesyłanie danych na serwer	315
12.5.1. Tworzenie serwera do obsługi żądań post	315
12.5.2. Tworzenie klienta do wysyłania żądań post	316
12.6. Przechwytywacze HTTP	319
12.7. Zdarzenia postępu	323

Podsumowanie 325

Rozdział 13. Interakcja z serwerami przy użyciu protokołu WebSocket 327

13.1. Porównanie protokołów HTTP i WebSocket 328

13.2. Wysyłanie danych z serwera Node do zwykłego klienta 330

13.3. Użycie gniazd WebSocket w klientach Angulara 333

13.3.1. Opakowywanie strumienia obserwowalnego w usługę 334

13.3.2. Komunikacja usługi Angulara z serwerem WebSocket 336

13.4. Część praktyczna: serwer Node z obsługą protokołu WebSocket 341

13.4.1. Uruchamianie aplikacji ngAuction w trybie programistycznym 344

13.4.2. Przegląd kodu serwera aplikacji ngAuction 344

13.4.3. Zmiany w kodzie klienta aplikacji ngAuction 351

Podsumowanie 357

Rozdział 14. Testowanie aplikacji Angulara 359

14.1. Testy jednostkowe 360

14.1.1. Wprowadzenie do frameworku Jasmine 361

14.1.2. Pisanie skryptu testowego dla klasy 363

14.2. Uruchamianie skryptów Jasmine za pomocą testera Karma 365

14.2.1. Plik konfiguracyjny Karmy 369

14.2.2. Testowanie w wielu przeglądarkach 370

14.3. Korzystanie z biblioteki testowej frameworku Angular 371

14.3.1. Testowanie komponentów 372

14.3.2. Testowanie usług 376

14.3.3. Testowanie komponentów wykorzystujących routing 379

14.4. Przeprowadzanie testów end-to-end przy użyciu biblioteki Protractor 382

14.4.1. Podstawy Protractora 383

14.4.2. Testy generowane przez CLI Angulara 386

14.4.3. Testowanie strony logowania 387

14.5. Część praktyczna: dodawanie testów e2e do aplikacji ngAuction	392
14.5.1. Przeprowadzanie testów e2e przepływu pracy dla wyszukiwania produktów	392
Podsumowanie	398
Rozdział 15. Utrzymywanie stanu aplikacji za pomocą biblioteki ngrx	399
15.1. Od sklepu ogólnospożywczego do architektury Redux	400
15.1.1. Co to jest Redux?	400
15.1.2. Dlaczego ważne jest przechowywanie stanu aplikacji w jednym miejscu?	402
15.2. Wprowadzenie do biblioteki ngrx	403
15.2.1. Poznajemy magazyn, akcje i reduktory	404
15.2.2. Poznajemy efekty i selektory	410
15.2.3. Refaktoryzacja aplikacji mediatora za pomocą ngrx	412
15.2.4. Monitorowanie stanu za pomocą rozszerzenia DevTools magazynu ngrx	420
15.2.5. Monitorowanie stanu routera	424
15.3. Używać ngrx albo nie używać	427
15.3.1. Porównanie ngrx z usługami Angulara	428
15.3.2. Problemy z mutacją stanu	429
15.3.3. Kod ngrx jest trudniejszy do odczytania	429
15.3.4. Krzywa uczenia się	430
15.3.5. Wnioski	430
15.4. Część praktyczna: stosowanie ngrx w aplikacji ngAuction	431
15.4.1. Dodawanie obsługi stanu routera do modułu aplikacji	433
15.4.2. Zarządzanie stanem w module głównym	434
15.4.3. Testy jednostkowe reduktorów ngrx	442
Podsumowanie	443
Angular 6, 7 i nie tylko	444
Dodatek A. Przegląd specyfikacji ECMAScript	447
Dodatek B. Podstawy TypeScriptu	481

Dodatek C. Korzystanie z menedżera pakietów npm 509

Dodatek D. Podstawy biblioteki RxJS 517

Skorowidz 539