

# React, TypeScript i Node Choi, David

O autorze

O recenzencie

Wstęp

**Część I. Jak zrozumieć TypeScript i poprawić swoją znajomość języka JavaScript**

## **Rozdział 1. Zrozumieć TypeScript**

- Wymagania techniczne
- Czym jest TypeScript?
- Dlaczego TypeScript jest niezbędny?
  - Typowanie dynamiczne a statyczne
  - Programowanie obiektowe
- Podsumowanie

## **Rozdział 2. Prezentacja języka TypeScript**

- Wymagania techniczne
- Czym są typy?
  - Jak działają typy?
- Wprowadzenie do typów języka TypeScript
  - Typ any
  - Typ unknown
  - Typy przecięć i unii
  - Typy literałów
  - Nazwy zastępcze typów
  - Typy wyników funkcji
  - Funkcje jako typy
  - Typ never
- Klasy i interfejsy
  - Klasy
  - Interfejsy
- Dziedziczenie
  - Klasy abstrakcyjne
  - Interfejsy
- Typy generyczne
- Prezentacja najnowszych możliwości języka i konfigurowania kompilatora
  - Łączenie opcjonalne
  - Scalanie wartości pustych
  - Konfigurowanie TypeScriptu
- Podsumowanie

**Rozdział 3. Tworzenie lepszych aplikacji dzięki użyciu możliwości wersji ES6+ języka JavaScript**

- Wymagania techniczne
- Poznanie rodzajów zmiennych w ES6 oraz zasięgów w języku JavaScript
- Poznanie funkcji strzałkowych
- Zmianie kontekstu this
- Rozproszenie, destrukuryzacja i reszta
  - Rozproszenie, Object.assign oraz Array.concat
  - Destrukuryzacja
  - Reszta
- Prezentacja wybranych funkcji tablicowych
  - find
  - filter
  - map
  - reduce
  - some oraz every
- Przedstawienie nowych typów kolekcji
  - Set
  - Map
- Przedstawienie słów kluczowych async i await
- Podsumowanie

## **Część II. Nauka tworzenia aplikacji jednostronicowych z użyciem frameworka React**

### **Rozdział 4. Przedstawienie koncepcji aplikacji jednostronicowych oraz ich realizacja z użyciem frameworka React**

- Wymagania techniczne
- Przedstawienie wcześniejszych sposobów tworzenia witryny WWW
- Cechy i zalety aplikacji jednostronicowych
- Jak React pomaga w tworzeniu aplikacji jednostronicowych
  - Atrybuty aplikacji Reacta
- Podsumowanie

### **Rozdział 5. Tworzenie aplikacji Reacta z wykorzystaniem hooków**

- Wymagania techniczne
- Wyjaśnienie ograniczeń i problemów związanych ze stosowaniem starych komponentów klasowych
  - Stan
  - Metody cyklu życia
- Prezentacja hooków Reacta i wyjaśnienie, dlaczego w stosunku do komponentów klasowych są one usprawnieniem
- Porównanie stosowania komponentów klasowych i hooków
  - Wielokrotne stosowanie kodu
  - Prostota
- Podsumowanie

### **Rozdział 6. Przygotowywanie projektu za pomocą create-react-app i testowanie go przy użyciu Jest**

- Wymagania techniczne
- Przedstawienie metod programowania aplikacji Reacta i systemu używanego do ich budowania
  - Narzędzia do zarządzania projektami
  - Transpilacja
  - Repozytoria kodu
- Testowanie aplikacji Reacta po stronie klienta
- Atrapy
  - Tworzenie atrapy z wykorzystaniem jest.fn
  - Tworzenie atrapy komponentów
- Prezentacja najpopularniejszych narzędzi oraz praktyk tworzenia aplikacji Reacta
  - Visual Studio Code
  - Prettier
  - Debugger Chrome
  - Alternatywne zintegrowane środowiska programistyczne
- Podsumowanie

## **Rozdział 7. Redux i React Router**

- Wymagania techniczne
- Zarządzanie stanem przy użyciu Reduxa
  - Reduktory i akcje
  - React Context
- Prezentacja frameworka React Router
- Podsumowanie

## **Część III. Tworzenie usług internetowych z użyciem Expressa i GraphQL-a**

### **Rozdział 8. Prezentacja tworzenia aplikacji serwerowych z wykorzystaniem Node.js i Expressa**

- Wymagania techniczne
- Wyjaśnienie sposobu działania środowiska Node
  - Pętla zdarzeń
- Prezentacja możliwości środowiska Node
  - Instalowanie Node
  - Tworzenie prostego serwera Node
  - Żądania i odpowiedzi
  - Trasowanie
  - Debugowanie
- Jak Express ułatwia pisanie rozwiązań przeznaczonych dla środowiska Node
- Przedstawienie możliwości frameworka Express
- Tworzenie internetowego API przy użyciu Expressa
- Podsumowanie

### **Rozdział 9. Czym jest GraphQL?**

- Wymagania techniczne
- Czym jest GraphQL?

- Schematy GraphQL
- Definicje typów i resolwery
- Zapytania, mutacje oraz subskrypcje
- Podsumowanie

## **Rozdział 10. Konfiguracja projektu Expressa z zależnościami od języków TypeScript i GraphQL**

- Wymagania techniczne
- Tworzenie projektu Expressa tworzonego w języku TypeScript
- Dodawanie do projektu GraphQL-a i jego zależności
- Prezentacja pakietów pomocniczych
- Podsumowanie

## **Rozdział 11. Czego się nauczysz - aplikacja internetowego forum**

- Analiza aplikacji, którą napiszemy - internetowego forum
- Analiza uwierzytelniania użytkowników forum
- Analiza zarządzania wątkami
- Analiza systemu punktacji wątków
- Podsumowanie

## **Rozdział 12. Tworzenie klienta Reacta na potrzeby aplikacji internetowego forum**

- Wymagania techniczne
- Tworzenie wstępnej wersji aplikacji Reacta
  - CSS Grid
  - Granice błędów
  - Warstwa usługi danych
  - Menu nawigacyjne
  - Komponenty związane z uwierzytelnianiem
  - Trasowanie i ekrany aplikacji
  - Ekran główny
  - Ekran wątku i jego wpisów
- Podsumowanie

## **Rozdział 13. Przygotowywanie stanu sesji przy użyciu Expressa i Redisa**

- Wymagania techniczne
- Czym jest stan sesji?
- Przedstawienie magazynu danych Redis
- Tworzenie stanu sesji z wykorzystaniem Expressa i Redisa
- Podsumowanie

## **Rozdział 14. Przygotowywanie Postgresa oraz warstwy repozytorium przy wykorzystaniu TypeORM**

- Wymagania techniczne
- Przygotowanie bazy danych Postgres

- Przedstawienie mechanizmów odwzorowań obiektowo-relacyjnych na przykładzie TypeORM
- Tworzenie warstwy repozytorium bazującej na Postgresie i TypeORM
- Podsumowanie

## **Rozdział 15. Dodawanie schematu GraphQL-a - część 1.**

- Wymagania techniczne
- Tworzenie definicji typów i resolverów dla serwerowego kodu GraphQL
  - System punktacji wątków
- Integracja mechanizmu uwierzytelniania z resolverami GraphQL-a
- Przygotowanie hooków Reacta do korzystania z serwera Apollo GraphQL
  - Ekran główny - komponent Main
  - Możliwości związane z uwierzytelnianiem
  - Ekran profilu użytkownika
- Podsumowanie

## **Rozdział 16. Dodawanie schematu GraphQL-a - część 2.**

- Komponent Thread i jego trasa
- System punktów
- Podsumowanie

## **Rozdział 17. Wdrażanie w chmurze AWS**

- Wymagania techniczne
- Konfiguracja Ubuntu w chmurze AWS
- Instalacja Redisa, Postgresa i Node w systemie Ubuntu
  - Instalacja serwera Redis
  - Instalacja Postgresa
  - Instalacja Node
- Konfiguracja i wdrażanie aplikacji na serwerze NGINX
  - Konfigurowanie projektu super-forum-client
  - Konfiguracja serwera NGINX
  - Rozwiązywanie problemów
- Podsumowanie