

Tworzenie aplikacji z wykorzystaniem NestJS

NODE/TS

Czas trwania: 4 dni

Jak budować nowoczesne aplikacje webowe w Node.js i TypeScript? Z pomocą przychodzi framework NestJS, mocno inspirowany Angularem, wprowadza w świat Node.js prostotę i porządek

Cele szkolenia

- Przekazanie praktycznej wiedzy pozwalającej tworzyć aplikacje serwerowe w oparciu o Node.js
- Poznanie dobrych praktyk, wzorców architektonicznych i narzędzi pozwalających na budowanie aplikacji, które będą skalowalne oraz łatwe w utrzymaniu i rozwoju
- Poznanie nowoczesnego frameworka NestJS i porównanie jego zalet na tle aplikacji opartej o Express
- Budowa REST API za pomocą NestJS wraz z automatycznie generowaną dokumentacją
- Połączenie z bazą danych SQL przy użyciu TypeORM
- Wykorzystanie i wzbogacenie wiedzy tak, by stworzyć skalowalną aplikację Node.js w NestJS
- Nauka jak pisać i uruchamiać testy jednostkowe oraz e2e

Zalety

- Szkolenie skupia się na tworzeniu aplikacji, które będą uruchamiane jako usługa lub w kontenerze Dockera
- Przekazana wiedza będzie pozwalała tworzyć oprogramowanie zgodne z ideą [The Twelve-Factor App](<https://12factor.net/pl>)

Dla kogo?

- Szkolenie jest przeznaczone dla programistów ze znajomością JavaScript w zakresie podstawowym, którzy chcą poznać nowoczesne metody budowy aplikacji serwerowych w technologii Node.js i TypeScript.
- Szkolenie jest specjalnie polecane dla programistów Angular, którzy chcieliby zacząć pisać backend, gdyż architektura NestJS jest mocno nim inspirowana.

Wymagania

- Od uczestników wymagana jest podstawowa znajomość JavaScript (ES2017)
- Zagadnienia, które należy znać: var, const, let, pętle, operacje warunkowe, funkcja, arrow function, klasa, ES Modules, Promise, async/await



Program

1. Wprowadzenie
 - a. Node.js
 - Jednowątkowa czy wielowątkowa architektura
 - Nieblokujące operacje wejścia/wyjścia
 - Moduły CommonJS
 - b. Node Package Manager (npm)
 - c. Asynchroniczność
 - Callback
 - Promise
 - async/await
 - Observable
 - d. TypeScript
 - ES Modules
 - Proces kompilacji TS do JS
 - Statyczne typowanie
 - Interfejsy
 - Dekoratory
 - Typy generyczne
2. Express
 - a. Budowa prostego serwera API
 - b. Routing
 - c. Architektura middleware
3. Architektura NestJS
 - a. Nest CLI
 - Generowanie nowej aplikacji
 - Generowanie komponentów aplikacji
 - b. Dependency Injection
 - c. Module
 - d. Controller
 - e. Providers
 - f. Service
 - g. Guard
 - h. Middleware
 - i. Custom Decorators
 - j. Pipes
 - k. Interceptors
 - l. Exception Filter
4. Budowa REST API
 - a. Routing
 - b. Autoryzacja
 - c. Walidacja
 - d. Upload plików



- e. Generowanie dokumentacji Swagger
 - f. Serwowanie HTML i plików statycznych
 - g. Konfiguracja
5. Bazy danych
- a. TypeORM
 - b. SQLite/MySQL
 - c. Migracje struktury bazy danych
6. Testowanie
- a. Testy e2e
 - Zapytania: GET, POST, DELETE...
 - Upload plików
 - b. Testy jednostkowe
 - Testowy moduł
 - Mockowanie serwisów
 - c. Automatyczne tworzenie danych testowych (fixtures)
 - Pliki i struktura katalogów
 - Uzupełnianie bazy testowymi rekordami
7. Inne konteksty wywołania
- a. CLI
 - uruchamianie elementów aplikacji z linii poleceń
 - uruchamianie zadań CRON
 - b. WebSocket - komunikacja real-time
 - c. RPC - Mikroserwisy
8. Build, deploy i monitoring aplikacji
- a. Budowa skryptów budujących z shell.js
 - b. Narzędzia do deploymentu
 - c. Monitoring i zarządzanie procesami z PM2

