



## Program szkolenia

# Node.js

### Szczegóły

---

**Kategoria:** Front-end

**Czas trwania:** 24h / 3 dni

**Termin:** ustalany indywidualnie

**Cena:** ustalana indywidualnie

## Opis

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

## Program

Poniżej przedstawiamy przykładowy program szkolenia, który może zostać zmodyfikowany zgodnie z oczekiwaniami oraz poziomem grupy szkoleniowej. Przed przygotowaniem docelowego programu szkolenia przeprowadzamy rozmowę techniczną, w której bierze udział trener oraz osoba techniczna lub cały zespół developerów reprezentujący klienta, w celu ustalenia szczegółów szkolenia.

### Dzień: 1 (8h)

#### 1. Wprowadzenie

- a. Node.js
  - i. Jednowątkowa czy wielowątkowa architektura
  - ii. Nieblokujące operacje wejścia/wyjścia
  - iii. Moduły CommonJS
- b. Node Package Manager (npm)
- c. Asynchroniczność
  - i. Callback
  - ii. Promise
  - iii. async/await
  - iv. Observable
- d. TypeScript
  - i. ES Modules
  - ii. Proces kompilacji TS do JS
  - iii. Statyczne typowanie
  - iv. Interfejsy
  - v. Dekoratory
  - vi. Typy generyczne

#### 2. Express

- a. Budowa prostego serwera API
- b. Routing
- c. Architektura middleware

### **3. Architektura NestJS**

- a. Nest CLI
  - i. Generowanie nowej aplikacji
  - ii. 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

## **Dzień: 2 (8h)**

### **1. Budowa REST API**

- a. Routing
- b. Autoryzacja
- c. Walidacja
- d. Upload plików
- e. Generowanie dokumentacji Swagger
- f. Serwowanie HTML i plików statycznych
- g. Konfiguracja

### **2. Bazy danych**

- a. TypeORM
- b. SQLite/MySQL
- c. Migracje struktury bazy danych

### **3. Testowanie**

- a. Testy e2e
  - i. Zapytania: GET, POST, DELETE..
  - ii. Upload plików
- b. Testy jednostkowe
  - i. Testowy moduł
  - ii. Mockowanie serwisów
- c. Automatyczne tworzenie danych testowych (fixtures)
  - i. Pliki i struktura katalogów
  - ii. Uzupełnianie bazy testowymi rekordami

## Dzień 3

### 1. Inne konteksty wywołania

- a. CLI
  - i. Uruchamianie elementów aplikacji z linii poleceń
  - ii. Uruchamianie zadań CRON
- b. WebSocket - komunikacja real-time
- c. RPC - Mikroserwisy

### 2. Build, deploy i monitoring aplikacji

- a. Budowa skryptów budujących z shell.js
- b. Narzędzia do deploymentu
- c. Monitoring i zarządzanie procesami z PM2

## Kontakt



### **Przemysław Wołosz**

Key Account Manager

infoShare Academy

+48 730 830 801

[przemyslaw.wolosz@infohareacademy.com](mailto:przemyslaw.wolosz@infohareacademy.com)