

# Tytuł szkolenia: Podstawy tworzenia aplikacji w Node.js – wersja rozszerzona

## Kod szkolenia: J-NODE-KOMPL

### Wprowadzenie

**Node.js** to wieloplatformowe środowisko uruchomieniowe **JavaScript** na licencji open-source, które umożliwia użycie kodu JavaScript bezpośrednio na komputerze, bez wykorzystania przeglądarki.

Node.js został stworzony przez Ryana Dahla na początku 2009 roku.

Node.js stał się technologią, po którą w ostatnich latach bardzo często sięgają start-upy oraz przedsiębiorstwa.

Dlaczego tak się stało? Co jest takiego pociągającego w Node.js?

**Szybkość działania, wysoka skalowalność, prędkość wytwarzania, społeczność** to najczęściej wymieniane aspekty wpływające na coraz większą popularność Node.js.

Dodatkowo dla przedsiębiorstw, bardzo istotny jest fakt, że Node.js jest obecnie częścią Linux Foundation, dzięki czemu ma długoterminowe wsparcie, co oznacza, że każde wydanie będzie wspierane przez okres 30 miesięcy.

Przedsiębiorstwa, które z powodzeniem wdrożyły Node.js:

- **PayPal**
- **Walmart**
- **LinkedIn**
- **NetFlix**
- **Capital One**

Każda z tych firm, ma swoją historię potwierdzającą ogromne korzyści wynikające ze stosowania platformy Node.js, dlatego nie czekaj dłużej i rozpocznij naukę Node.js!

### Adresaci szkolenia

Adresatami szkolenia są programiści JavaScript, pragnący poznać platformę Node.js.

Minimalne wymagania wstępne:

- Umiejętność programowania w JavaScript.
- Podstawowa znajomość ES2015

### Cel szkolenia

Szkolenie koncentruje się na przybliżeniu użytkownikowi platformy Node.js, aby mógł swobodnie rozpocząć przygodę z budowaniem aplikacji przy użyciu tego narzędzia.

Po zakończeniu szkolenia aktywny uczestnik potrafi:

- opisać, czym jest **Node.js**
- posługiwać się narzędziem **npm**
- stosować asynchroniczne techniki programowania w **JavaScript** oraz **Node.js**
- stworzyć prosty serwer HTTP przy użyciu **Express**
- połączyć swoją aplikację z bazą danych
- pobrać zasoby z zewnętrznych źródeł
- stworzyć strukturę projektu
- wykorzystać oprogramowanie pośredniczące Passport.js do uwierzytelnienia użytkowników z pomocą **Redis-a**
- napisać testy jednostkowe z użyciem frameworka **Mocha, Chai** oraz **Sinon**
- debugować aplikację Node.js
- zabezpieczyć aplikację przed częstymi problemami dotyczącymi bezpieczeństwa
- deployować aplikację Node.js z użyciem **Dockera**
- monitorować aplikację Node.js

Domyślnie w czasie szkolenia uczestnicy korzystają ze środowiska programistycznego WebStorm na Windowsie.

## Czas i forma szkolenia

- 35 godzin (5 dni x 7 godzin), w tym wykłady i warsztaty praktyczne.

## Plan szkolenia

- 1. Poznajemy Node.JS**
  - a. Historia Node.js
  - b. Główne funkcje Node.js
  - c. Jak działa Node.js?
  - d. Co możemy zrobić z użyciem Node.js?
  - e. Instalujemy Node.js
  - f. Architektura Node.js
  - g. Korzystamy z Node.js po przez terminal (CLI i REPL)
  - h. Tworzymy prostą aplikację w Node.js
- 2. Poznajemy system modułów występujący w Node.js**
  - a. Standard CommonJs
  - b. Jak działa require?
  - c. Moduły ES6
  - d. Wzorce definicji modułów
- 3. Poznajemy NPM**
  - a. NPM w pigułce
  - b. Tworzony nowy projekt
  - c. Poznajemy package.json
  - d. Zarządzenie zależnościami projektu
  - e. Skrypty npm
- 4. Zrozumieć asynchroniczne programowanie**
  - a. Problem z synchronicznym programowaniem
  - b. Poznajmy pętlę zdarzeń
  - c. Callbacks
  - d. Promise
  - e. Async/Await
  - f. EventEmitter
- 5. Wbudowane moduły Node.js**
  - a. Buffer
  - b. Stream
  - c. File System
  - d. OS
  - e. Path
  - f. Net
  - g. DNS
- 6. Piszemy prosty serwer HTTP**
  - a. Serwer przy użyciu modułu http
  - b. Poznajemy frameworkExpress
    - i. Tworzymy serwer obsługujący Http z użyciem Express
    - ii. Praca z oprogramowaniem pośredniczącym
    - iii. Obsługa błędów
    - iv. Silnik szablonów
    - v. Parametry i routing
    - vi. Debugowanie Express
  - c. Korzystanie z zewnętrznych źródeł przy pomocy modułu request
- 7. Struktura projektu Node.js**
- 8. Techniki debugowania Node.js**
  - a. Przy użyciu modułu debug

- b. Przy użyciu wbudowanego debuggera
- c. Przy użyciu Chrome DevTools
- 9. **Używamy Swagger-a do pracy z REST API**
- 10. **Piszemy prosty czat z użyciem Socket.io**
- 11. **Poznajemy możliwości używania bazy danych**
  - a. Łączymy aplikację Node.js z postgresql
  - b. Korzystamy z Sequelize będącego ORM dla Node.js
- 12. **Uwierzytelniania za pomocą Passport.js z użyciem Redis-a**
  - a. Krótkie wprowadzenie do Redisa
  - b. Przygotowujemy przykładową aplikację
  - c. Konfigurujemy Passport.js dla Node.js
  - d. Dodajemy chronione trasy
- 13. **Piszemy testy jednostkowe z użyciem Mocha, Chai oraz Sinon**
  - a. Wprowadzenie do pisania testów
  - b. Poznajemy moduł asercji Chai
  - c. Poznajemy moduł Sinon do tworzenia szpiegów i moków
  - d. Piszemy test jednostkowy z użyciem Mocha, Chai oraz Sinon
  - e. Dodajemy raport pokrycia kodu
- 14. **Bezpieczeństwo aplikacji**
  - a. Zabezpiecz styl kodowania
  - b. Uruchom w bezpieczny sposób aplikację na produkcji
  - c. Narzędzia wspomagające wyszukiwanie niebezpiecznych modułów
- 15. **Deploy aplikacji Node.js używając Dockera**
  - a. Krótkie wprowadzenie do Dockera
  - b. Stworzenie obrazu Docker-a dla aplikacji Node.js
  - c. Deploy obrazu dockera na dostawcę PaaS
- 16. **Monitorowanie aplikacji Node.js**
  - a. Wyjaśnienie potrzeby monitorowania aplikacji Node.js
  - b. Radzenie sobie z błędami
  - c. Monitorowanie z użyciem narzędzi pm2, nodetool