

## Program szkolenia:

# Node.JS - Tworzenie reaktywnych punktów dostępu (API gateway) i REST API

## Informacje:

<b>Nazwa:</b>	<b>Node.JS - Tworzenie reaktywnych punktów dostępu (API gateway) i REST API</b>
<b>Kod:</b>	<b>node-gateway</b>
<b>Kategoria:</b>	Node.js
<b>Odbiorcy:</b>	developerzy, architekci
<b>Czas trwania:</b>	3 dni
<b>Forma:</b>	80% Warsztaty / 20% Wykłady

Szkolenie przeznaczone dla programistów Back-End chcących tworzyć reaktywne punkty dostępu i systemy serwerowe przy użyciu JavaScript oraz serwera NodeJS.

Szkolenie wprowadza w specyfikę programowania reaktywnego, bez znajomości której niemożliwe jest tworzenie wydajnych aplikacji na platformę NodeJS. Uczestnicy w praktycznych przykładach poznają najlepsze praktyki oraz metody unikania najczęstszych błędów.

W miarę potrzeb zostaną też wdrożeni w bardziej zaawansowane części języka JavaScript oraz sposoby na klastrowanie i zabezpieczanie aplikacji.

## Zalety szkolenia:

- Najlepsze praktyki
- Sprawdzone wzorce
- Zorientowanie na wydajność

## Szczegółowy program:

### 1. NodeJS Concepts

1.1. Modules

1.2. Can I Use for NodeJS

1.2.1. node.green

1.3. Buffers/Streams

1.4. EventEmitter

1.5. Memory Model

### 2. Express JS

2.1. Routing

2.2. Handlers

2.3. Middlewares

2.4. Request/Response

2.5. Static Content

2.6. Parsing Body

### 3. Functional Concepts

3.1. Map/reduce/forEach/filter

3.2. Higher Order Function

3.3. Partial Application

### 4. Threading / Async

4.1. Callbacks

4.2. Promises

4.2.1. Promisify Node APIs (util.promisify)

4.3. Async JS

#### 4.4. Event Loop

##### 4.4.1. Stack Machine

##### 4.4.2. Queues

##### 4.4.3. Threading Model

### 5. IO in NodeJS

#### 5.1. Synchronous vs asynchronous

#### 5.2. Events

#### 5.3. Streams

### 6. Application Security

#### 6.1. Passport

#### 6.2. Bcrypt

#### 6.3. JWT

#### 6.4. Sessions/Cookies

### 7. Testing

#### 7.1. Unit - Jest

#### 7.2. Async tests

#### 7.3. Component - Supertest

#### 7.4. Testable code

##### 7.4.1. Dependency Injection

### 8. Utilities

#### 8.1. Lodash

#### 8.2. Logging

##### 8.2.1. Access logs

##### 8.2.2. Development Logging

##### 8.2.3. Production Logging

8.3. Debugging

8.3.1. Chrome Debugger

## 9. Deployment/Infrastructure

9.1. PM2

9.1.1. Load Balancing/Cluster

9.2. Apache Proxy

## 10. JS Refresher

10.1. Functions

10.2. Context (this)

10.3. Closures

10.4. Objects

10.5. Classes

10.5.1. Prototypes

10.6. Visibility

10.7. Arrow functions

10.8. Destructuring

10.9. Callbacks