Node.js for JavaScript Developers

CODICE	**DT0119**
DURATA	**3 gg**
PREZZO	**1.200,00 €**
EXAM	

DESCRIZIONE

Node.js è una piattaforma per creare facilmente applicazioni di rete veloci e scalabili. Scopri come utilizzare il modello I / O basato su eventi e non bloccante per creare applicazioni in tempo reale ad alta intensità di dati che possono essere eseguite su dispositivi distribuiti.

Esploreremo le scelte di progettazione che rendono unico Node.js , come questo cambia il modo in cui le applicazioni sono costruite e come i sistemi di applicazioni funzionano in modo più efficace in questo modello. Scopri come creare un codice modulare robusto, espressivo e chiaro. Comprendi quando utilizzare callback, emettitori di eventi e stream. Utilizzare i flussi per manipolare facilmente i dati che sarebbero impossibili da adattare nella memoria dell'applicazione. Acquisire fiducia nella gestione efficace degli errori per garantire l'affidabilità del runtime. Il corso prevede approfonditi esercizi di laboratorio per rafforzare i concetti e le tecniche trattati.

TARGET

- Web application developers
- Java developers

PREREQUISTI

Esperienza nella programmazione JavaScript

CONTENUTI

Intro to Node.js

- RAM vs. I/O latency
- Blocking vs. Non-Blocking
- Event-driven Programming
- Event Loop
- Blocking The Event Loop

- Node.js Philosophy

Node.js Platform Setup

- Download and Install
- Node REPL
- First Hello World

Modules and npm

- Anatomy of a module
- Private code
- Accessing and using modules
- npm commands
- package.json

The Callback Pattern

- What are callbacks
- Callback-last
- Error-first

Events

- When to use Event Emitters
- Binding Functions to Events
- Event Requests
- Event Listening

Error Handling

- Callbacks: Error-first
- Errors in Event Emitters
- Uncaught Exceptions
- Using Domains

Buffers

- Why Buffers exist
- Creating Buffers
- Reading and Writing Buffers
- Manipulating Buffers

Streams

- What are streams
- Read and Write Stream API
- Flow Control
- Piping
- Duplex Stream
- Transform Stream

Express.js

- Intro and Installing Express.js
- Building a Hello Express application
- Creating routes
- Rendering Layouts
- Using templates
- Adding partials
- Using locals and conditional templates
- Modularizing routes

Socket.io

- Listening for
- Broadcasting
- Answering questions

Connecting to Databases

- No SQL and Document Stores
- Relational DBs
- Configuration and platform setup
- CRUD Operations