

Sela.

NodeJS

NodeJS

college@sela.co.il

03-6176666





NodeJS

NodeJS - Version: 4

 2 days Course

Description:

It seems like the world has gone mad. For years we have been struggling with JavaScript on the client side, and now all of the sudden people can't have enough of it on the server side. They claim it is "fun". This is not a fringe phenomenon: Node.js is being used by giants like eBay, LinkedIn and Microsoft. In this tutorial we will learn to embrace the Zen of Node.js.

Intended audience:

This course is intended for professional Web developers, team leaders and architects who have knowledge in JavaScript. Members of the audience should have a minimum of one to two years of experience developing Web-based applications.

Prerequisites:

Basic JavaScript skills.
Web Development background

Objectives:

Topics:

Asynchronous JavaScript Everywhere



- Introduction - What is Node.js.
- The importance of being asynchronous
- Node.js Event loop
- The Node.js process

File System

- The FS Module.
- Reading Directories
- Reading Files
- Streams

Building servers

- Creating servers with HTTP
- HTTP streaming
- The Http server class
- Parsing requests

Building APIs using modules, events and packages

- CommonJS Modules
- The global object
- Directories as Modules
- npm Packages
- Managing Dependencies
- npm scripts

Async Patterns



- Asynchronous Javascript
- Asynchronous Control Flow with Callbacks
- Promises
- Async/Await
- The EventEmitter API
- Third Party Async packages

ExpressJS

- REST
- Introduction to ExpressJS
- Routing
- Responding
- Configuration
- Views
- Middlewares
- Receiving Data
- Error Handling

Working With Data

- MongoDB
- Mongoose ODM
- Mongoose Schema
- Mongoose Model
- Querying with Mongoose

Authentication With Passport and JWT

- Stateful vs. Stateless Authentication



- OAuth2
- Passport
- JWT – JSON Web Tokens

Hosting Standalone Node

- Why do we need hosts
- Forever
- PM2
- Node on Windows
- Node as a Windows Service
- IISNode

Hosting Complex Node Architectures with Docker

- What is Docker
- Docker CLI
- Docker File system & Volumes
- Docker Files
- Containers Communication
- Docker Compose

Best Practices

- NPM
- Best Practices
- Node.js API design
- Error Handling
- Debugging