

Real-time web applications with Node.js and Express.js

Course 3376 – 40 Hours

Overview

Node.js is a software system designed for writing highly scalable Internet applications, notably web servers. Programs are written in JavaScript, using event-driven, asynchronous I/O to minimize overhead and maximize scalability. Unlike most JavaScript programs, it is not executed in a web browser, but is instead a server-side JavaScript application . Node.js consists of Google's V8 JavaScript engine plus several built-in libraries.

On Completion, Delegates will be able to

In this seminar you'll learn the fundamentals of Node.js and how to use Node.js to build lightweight, real-time full stack web-applications with Node and the Express framework. Participants will also learn how to interface Node.js with back-end databases and discover how to use several of the leading external modules for Node.js

Who Should Attend

This seminar is for web developers experienced in OOP with basic knowledge of JavaScript, who wish to develop cross-platform JavaScript applications.

Prerequisites

Experience in OOP with basic knowledge of JavaScript

Course Contents

Introduction to Node.js

- What is Node and what is it not
- Node.js Features
- Our first Node.js script: Hello World
- Hello Server: Building a web server in Node.js
- Debugging node applications

Building your Stack

- Pulling in other libraries
- Building custom libraries
- Asynchronicity and callbacks
- Blocking vs. non-blocking I/O
- Working within the event loop



Modular JavaScript with Node.js

- Writing Modular JavaScript with Node.js
- Core Modules
- Installing Packages
- Publishing packages

Avoiding common pitfalls with Async.js

- Introducing the Async problem
- Async.js Library to the rescue
- Collections
- Flow Controllers

Working with the file system

- Sync and Async operations
- Files manipulations
- Folder manipulations
- Putting the file-system module together Async.js

Data Access 01 – mySql

- Installing mySql node package
- Simple db connection
- CRUD example
- Putting together Async with mySql operations

Command-line interfaces

- The built-in REPL
- Custom REPL
- Using external libraries
- Receive command-line arguments
- Build command-line tools

Unit-Testing Node Applications

- Introduction to unit testing
- Testing with Mocha and Should
- Suits, specs & Reporters
- Testing Synchronous code
- Testing Asynchronous code

Building Web applications with the Express Framework

- Introduction to Express, installation and basic setup
- Application configuration
- Routing
- Views and Templating options
- Persistence with Cookies, In-Memory Sessions and session-stores.
- Authenticating users with passport local



- Social Auth with Passport.js

Data Access 02 – mongoDB

- Tooling up – installing mongo, clients and drivers.
- Mongoose Schemas
- CRUD operations
- Single Page Applications with Express, Mongoose and Angular.js

Real-time communication

- Introduction to real-time applications
- Listen & emit
- Readable streams – streaming chunked data
- Piping Readable streams to Writable streams
- Sockets on the Server and the Client
- Build a chat application