

Node.js Hands-on Workshop

OVERVIEW

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient.

A Node.js along with the Express framework help quickly create backend web applications. A popular choice of database is MongoDB.

PREREQUISITES

- Working knowledge of HTML & CSS
- Good knowledge of JavaScript
- Bootstrap knowledge is a plus, but not necessary
- Very basic algorithms and data structures knowledge (traversing and using arrays, using objects in JavaScript)
- Good knowledge of web application concepts

TOPICS COVERED

1. Introduction to Node.js
 - a. What is Node.js?
 - b. Node.js architecture
 - c. Event loop
 - d. Blocking vs Non-blocking
 - e. NPM
 - f. Installing Node.js
 - g. Node REPL
2. Node Module System
 - a. Global object
 - b. Custom Modules
 - i. Creating a custom module
 - ii. Loading a custom module
 - iii. Using a custom module
 - c. Core Modules
 - i. Path module
 - ii. OS Module
 - iii. File System Module
 - iv. Events Module
 - v. Http Module
 - d. Third-Party Modules
3. Node Package Manager (NPM)
 - a. Introduction
 - b. Package.json
 - c. Installing a package
 - d. Using a package
 - e. Viewing Registry info for a package
 - f. Installing a specific version of a package
 - g. Updating local packages
 - h. Dev dependencies
 - i. Uninstalling a package
 - j. Working with global packages
4. Introduction to Express
 - a. What is Express?
 - b. Installing Express
 - c. Express middleware
 - i. Custom middleware
 - ii. Built-in middleware
 1. JSON parser
 2. URL Encode parser
 3. Serving static files
 - iii. Third-party middleware
 1. Morgan

- d. Template engines
- e. Express App Generator
- 5. Building RESTful APIs using Express
 - a. Introduction to RESTful services
 - b. Routing
 - c. Route parameters
 - d. Query parameters
 - e. Handling HTTP GET requests
 - f. Handling HTTP POST requests
 - g. Handling HTTP PUT requests
 - h. Handling HTTP DELETE requests
 - i. Testing API endpoints using Postman
 - j. Refactoring routes into separate modules
 - i. Express Router
- 6. Asynchronous JavaScript
 - a. Synchronous vs Asynchronous code
 - b. Patterns for Asynchronous code
 - c. Callbacks
 - d. Callback hell
 - e. Promises
 - f. Async and Await – Overview
- 7. Introduction to MongoDB
 - a. NoSQL – What & Why?
 - b. Overview of MongoDB
 - c. Setting up MongoDB
 - d. JSON
 - e. BSON
 - f. CRUD using Mongo Shell
 - g. MongoDB Node.js driver
- 8. Mongoose
 - a. Introduction
 - b. Connecting to MongoDB
 - c. Schema and schema types
 - d. Model
 - e. Document
 - f. Querying documents
 - g. Validation
- 9. ES6
 - a. Overview
 - b. Array helper methods
 - c. Const & let
 - d. Template literals
 - e. Arrow functions
 - f. Default function arguments
 - g. Rest & Spread

- h. Destructuring – Array & Object
- i. Classes & Inheritance
- j. Promises
- k. Fetch

DAYWISE BREAKUP OF TOPICS

Sl. No.	Topic	Day
1	Introduction to Node.js	1
2	Node Module System	1
3	Node Package Manager (NPM)	1
4	Introduction to Express	2
5	Building RESTful APIs using Express	2
6	Asynchronous JavaScript	2
7	Introduction to MongoDB	3
8	Mongoose	3
9	ES6	3