

NodeJS

Duration: 5 days

Objectives:

- Learn why server-side JavaScript is useful.
- Learn the design and architecture of Node.js applications.
- Build an HTTP server using the core modules in Node.js.
- Developing web applications with ExpressJS.
- Developing scalable REST endpoints with ExpressJS.
- Gain proficiency in utilizing NPM.

Prerequisites:

- A prerequisite for participation is proficiency in programming.
- Familiarity with Web Applications is essential for all participants.
- Basic understanding of JavaScript is a requirement for participants.

Prerequisites tutorial

<https://developer.mozilla.org/en-US/docs/Web/JavaScript>

Contents

Introduction

- What is Node.js?
- Why Node.js?
- Node Modules
- Node Built-in packages.
- Creating your own package.
- Import your own Package.
- NPM (Node Package Manager)
- Local and Global Packages
- Nodejs architecture

Asynchronous Programming

- What is Asynchronous Programming in JavaScript?

- JavaScript EventLoop
- Call Stack
- Callbacks
- Callback Hell
- Promises
- Promise Chaining
- Async and Await
- Event Emitter

Built-In Libraries

- Working with the Operating System
- Working with the File System
- Console and Utilities
- Debugging

File System

- Synchronous and Asynchronous IO
- Listing and reading directories and files.
- Writing and updating to files
- File System Watcher

Stream and Events

- What are Streams in Node.js
- Reading and Writing Streams
- Understanding and using EventEmitter class
- Emitting Events
- Responding to events

NPM

- What Is a Package
- Typical NPM Usage
- NPM Help
- Creating a Package.json
- Setting Defaults
- Installing Packages
- Listing Installed Packages
- Installing Global Packages
- Removing a Package
- Installing Specific Versions
- The NPM Registry
- Searching for Packages
- Pruning
- Using Simple Scripts
- Publishing Your Package

HTTP and REST

- The HTTP protocol
- Node HTTP Module
- Making a request and response mechanism
- Build a web server
- Serving to files
- Serving to Json Data
- Collection post data from web pages
- Scaling the application

ExpressJS

- What is ExpressJS
- The model-view-controller pattern
- Creating a simple HTTPServer using ExpressJS
- Building a front-end controller
- Routing in Express
- Templating in Express
- Templating using JADE
- Defining routes
- Middewares

ExpressJS and MongoDB

- What is MongoDB
- MongoDB Advantages
- Documents Collections
- Mongoose ODM
- Creating Schemas and Models using mongoose
- CRUD Operations in MongoDB using Mongoose and Express

Working with Web Sockets

- Creating a web socket server
- Broadcasting message with web sockets
- Creating web sockets with Sockets.io.
- Realtime interaction with Socket.io

ExpressJS and REST

- Understanding the principles of REST
- Richardson Maturity Model
- Creating a RESTful Web Service using Node Express
- Data Exchange Formats JSON vs. XML
- Monolith Architecture
- Microservice Architecture
- Securing REST services

Building Scalable application using NodeJS

- Worker Threads
- Child Process
- Cluster Modules
- Cloud Deployment