

Node JS *Curriculum* {20Days}

Office Location

DreamTech Labs
Jalandhar

Office Hours

9:30-18:00



Course Overview

Node.js is an open source, cross-platform framework running on Mac OSX, Windows, and Linux. Node.js course has been designed to help developers understand and build web applications with the help of JavaScript. Node.js is an event-driven, server-side JavaScript environment that runs JavaScript using the V8 engine developed by Google. The course focuses on various important concepts of Node.js and provides hands-on experience in building HTTP server with Node.js, working with file system, buffers, stream, events, and multi-processing in Node.js. The training covers Express.js, which is a powerful and flexible web application framework that provides a robust set of features for web and mobile applications.

OBJECTIVES

This course will enable you to:

- Understand the Node.js framework
- Work with Node Projects
- Work with shrink-wrap to lock the node module versions
- Learn asynchronous programming
- Build HTTP Server with Node.js using HTTP APIs
- Understand file systems in Node.js
- Understand Buffers, Streams and Events
- Comprehend multi-processing in Node.js
- Master Express.js
- Learn how Express.js connects with MongoDB and SQLite
- Build chat application using Socket.io
- Work with Gulp and Grunt
- Understand unit and End to End (e2e) testing

PREREQUISITES

The prerequisites to learn Node.js course includes basic knowledge of JavaScript and HTML.

Course Materials

Introduction to Node.JS

- * Introduction
- * The Node.js framework
- * Installing Node.js
- * Using Node.js to execute scripts

Building a HTTP Server with Node.JS using HTTP APIs

- * The HTTP Protocol.
- * Building an HTTP server.
- * Rendering a response.
- * Processing query strings.
- * Processing posted data.

Buffers, Streams and Events

- * Using binary for buffer data.
- * Flowing v/s Non flowing streams.
- * Streaming I/O from files.
- * Processing streams asynchronously.

Socket IO, Front End, and A chat App

- * Getting Started.
- * Adding socket IO to App.
- * Working with time.
- * Show Messages Time in Chat App