

MongoDB Express.js React Node.js STACK



Coding Club
IIT Guwahati



Powered by
HASURA

Contents

- Node.js - What and Why ?
- Introduction
- Set-up

02

- Client-Server architecture and Requests and responses
- Node Package Manager
- Framework
- MVC (Model View Control)
- API s

03

- View Engines
- Middleware
- Databases
- Router Packages
- Testing

04

- Wrap up

05

Node.js

What is NodeJS?

- We use HTML, CSS, JavaScript for making webpages, but when we run these files, they are usually run by our browser. JavaScript files cannot be run by our system directly and needs some kind of tool to run it.
- NodeJS is such a Server-Side JavaScript runtime environment. It compiles the JavaScript code into native machine code.

Why NodeJS?

- It provides good throughput and higher consistency and is thus heavily used by companies for building large scalable single page applications, video streaming applications and web applications.
- NPM (Node package Manager) comes with 800,000+ libraries that can be used for different purposes.

Prerequisites :-

- There is no strict prerequisite for this particular course, although it is advised that the reader learns a bit of JavaScript concepts before tackling NodeJS.
- Even without any JavaScript knowledge, this course can be easily followed along with other JavaScript courses side-by-side if you have some programming background.

Setting things up :

- Installing NodeJS is a very simple procedure. You can download the stable version of NodeJS from the [NodeJS official site](#).
- NodeJS official site. You can also refer the [following video](#) for setting up the editor and also get a intro to Node.

NodeJS Intro

- File systems and Modules: <https://youtu.be/OIBIXYLJjsI>
- Node.js Basics:
File System: https://www.w3schools.com/nodejs/nodejs_filesystem.asp
Modules: <https://nodejs.org/api/modules.html#modules-commonjs-modules>

Client-Server architecture and Requests and responses

Clients & Servers : <https://youtu.be/-HPZ1leCV8k>

Computer networking basics- Clients and servers : <http://shorturl.at/eflW5>

Requests and responses : <https://youtu.be/DQD00NAUPNk>

NPM (Node Package Manager)

The Node Package manager is one of the most important core concepts to learn in NodeJS. NPM comes with 800,000+ libraries. These libraries can be installed using the npm command and speeds up the application development process.

[NPM Tutorial](#)

[NPM Reading material](#)

Framework

Like other server side languages, NodeJS also provides a number of frameworks to make your work easier. Some popular NodeJS frameworks include **Express.js**, Meteor.js, Adonis.js etc.

In this course we will follow through with Express

[Express Youtube tutorial](#)

[Reading material](#)

MVC (Model View Control) frameworks

These are libraries that can be included alongside JavaScript to help structure the code-base and separate the concerns of an application in three parts: Model (representing the data), View (user interface of the application), Controller (Handling inputs).

[Blog](#)

[Reading Material](#)

[Events Module](#)

APIs

API stands for Application Programming Interface which basically allows two applications to talk to each other. To get a better understanding, refer to the following videos:

- [Intro to APIs](#)
- [API for beginners- Full Course](#)
- [REST API using node and HTTP module](#)
- [Creating a Fake REST API using JSON server](#)
- [Working with JSON in NodeJS](#)

View Engines

View engine works inside the application for rendering HTML page to the browser or to the user. It can contain HTML, tags, server controls and some programming language.

[View engines- Net Ninja tutorial](#)
[Reading material](#)

Middleware

A middleware in Node.js is a function that will have all the access for requesting an object, responding to an object, and moving to the next middleware function in the application request response cycle.

[Net Ninja tutorial](#)
[Reading material](#)

Databases

Depending on the type of application or requirement we can choose any database and configure it with NodeJS.

[Youtube video](#)
[Reading Material](#)

Router Packages

Router packages like express router are used to split routes (path a data packet travels) into separate files.

[Using routes](#)
[Express Router and MVC](#)
[Reading material](#)
[Blog- NodeJS router and routes](#)

Testing

Any application requires proper testing before it is released into the market. So, you should learn some of the testing frameworks and libraries for NodeJS applications. Below are links to some testing tutorials for NodeJS using different Frameworks:

[Testing with JEST](#)
[Unit testing with Chai and Mocha](#)
[E2E testing, puppeteer and Node.js](#)

Wrap up

- So far we have covered most of the basics in Node and some more. But there are many features and applications of node that were not discussed here in this course .
- It is also not possible to cover everything that node has to offer and thus, the rest is left up to the reader's discretion.
- After clearing your basics with this course, you should look for more applications of node specific to the area where you are interested in developing, be it a blog site, or educational site, or other OTT etc.
- To put everything you have learned so far, try doing a project based on the same.
- Refer to the following video for a NodeJS based project building demonstration:

NodeJS Project- Banking Application

Complete CRUD Application (MongoDB also used)

So, we will wrap up here. That's all for this short crash course on NodeJS.

Submission Link for Week 2 : <https://forms.gle/MTvPuZCn4RMvb4DU7>