Get started with Node.JS

Learning Objective

Discover Node.js and its capabilities and understand why it makes creating server-side applications easy and fast. Learn how to setup Node.js and write your first script before diving into synchronous and asynchronous programming and understanding the all-important event loop and non-blocking I/O.

Topics

- Introduction to Node.js
- Applications of Node.js and installation
- Writing your first Node.js Script
- Synchronous and Asynchronous programming
- Under the hood - understanding the event loop and Non-Blocking I/O

Interactive Node with REPL

Learning Objective

Understand what is Node.js REPL and its commands. Learn all about global and local objects in Node.js.

Topics

- REPL and REPL Commands
- Node.js CLI Commands
- Global and Local Objects

Modular Programming and NPM

Learning Objectives

Learn all about modular programming with Node.js and NPM. Discover the power of modules as you write your own module. Then learn all about using npm to set up and initialize a project before understanding package.json, local and global packages, using npx, and finally publishing a package on npm.

Topics

- Introduction to Modules
- Process and OS Modules
- Writing Your Own Module
- Introducing NPM
- Initializing a Project Using npm init
- Understanding package.json
- Installing and using packages from npm
- Local vs.Global Install
- Using NPX
- Publishing Package on npm

Hands-on

- Write your own module

Introduction to ECMAScript Modules

Learning Objectives

Node.js offers experimental support for ECMAScript modules, out of the box. Learn all about this feature and use it in your projects without the need of a third-party compiler such as Babel. You will learn to write and import an ECMAScript module as you learn about its syntax and uses.

Topics

- ECMAScript Modules Versus CommonJS Modules
- Enabling support
- Writing an ECMAScript Module
- Importing and Using an ECMAScript Module
- ES6 syntax in detail

Hands-on

- Importing and Using an ECMAScript Module

File System and Streams

Learning Objectives

Node.js allows you to work with the file system. This module covers the all-important 'fs' module as you learn to work with files and directories. You will learn to read and write files both synchronously and asynchronously.

Topics

- File System Modeling in Node.js
- Directory and Path Resolution
- Reading Files Synchronously and Asynchronously
- Writing Files Synchronously and Asynchronously
- Directory operations

Hands-on

- Learn how to write a file using Sync and Async.
- Learn all about Directory Operations.

Streams

Learning Objectives

Discover the Streams API in Node.js and its use case in the form of reading and writing files. You will also learn about transform streams.

Topics

- Understanding Streams
- Reading a File Using Readable Stream
- Writing a file using Writable Streams
- Transform Streams
- Streams vs Files

Hands-on

- Learn how to work with transform streams

Events

Learning Objectives

Learn to use the events module to create your own events and emit them. Learn to write an event handler for an event raised by your own created custom events.

Topics

- Event Emitter
- Handling Events

Network I/O

Learning Objectives

Understand the purpose of the net module and how to use it in a node application. Learn to create a server that listens when the client connects to it. Also, learn to create a client that communicates with the server and is able to exchange messages.

Topics

- Introduction to the Net module
- Creating a TCP Server and Listener
- Creating a command-line chatbot

Web Servers and More

Learning Objectives

Learn all about building servers using the HTTP module. You will also learn about handling incoming requests and sending out responses as you build a web server that serves a static website. Learn all about HTTPS and HTTP/2 and learn to handle Cross-Origin Resource Sharing (CORS).

Topics

- Web Server: Web Application Architecture
- HTTP module
- Making HTTP requests
- Serve JSON as a response
- Serve HTML as a response
- Building a basic web server with routes and streams: Serve a static website
- Using HTTPS
- Using HTTP/2
- Understanding CORS

Hands-on

- Learn how to make HTTP Requests
- You will learn serving JSON as a response
- You will learn serving HTML as a response
- You will learn serving a static website

Debugging in NodeJS

Learning Objectives

Understand debugging and its importance in an application development environment. Learn how to debug a Nodejs application with an inspect flag.

Topics

- Introduction to debugging
- Debug node js apps

Cluster and Worker Threads

Learning Objectives

Discover threads and the worker thread API in Node.js. Learn to write CPU intensive code using the worker threads and understand the need for clusters for scaling up a Node.js app.

Topics

- Introduction to the Worker Threads API
- Handling compute-intensive tasks using Worker Threads
- Node.js on multi-core CPUs using Cluster

Hands-on

- You will be handling compute-intensive tasks using Worker Threads

Introduction to Express

Learning Objectives

Learn about the need for an application framework when building Node.js apps. Build a basic Node.js application framework. Discover Express and the benefits it brings to application development. Learn about real-world customers who bank on Express for their production apps.

Topics

- What is a Node.js web application framework?
- BYOF - Build Your Own Framework
- Introducing Express
- Who uses Express?

Hands-on

- You will build your own framework

Hello Express

Learning Objectives

Build the very first Express app. Learn to add a route handler for POST requests in your Express application, and learn to use the express-static middleware to serve static assets.

Topics

- Build your first Express app
- Learn adding a route handler

- Serving Static Assets and HTML files

Hands-on

- You will build your first Express app
- Add multiple route handlers to the app
- Serve a static signup page with its stylesheet

Rendering

Learning Objectives

Learn and configure a view engine to render dynamic Pug templates. Next, learn to build a simple dashboard page using Pug. Finally, build and include a partial using Pug.

Topics

- View Engine – Pug
- Building the Dashboard
- Building the Post Card

Hands-on

- You will be rendering a dynamic signup page using Pug

Middleware

Learning Objectives

Discover middleware and its role in an Express application. Learn to handle and parse incoming requests such as form data using middleware. Next, learn to use Morgan, a third-party middleware that helps you log requests. Understand how to implement sessions using the express session middleware. Finally, learn to write your own middleware function and implement response data compression.

Topics

- What is middleware?

- Parsing incoming requests
- Logging with Morgan
- Using the express-session middleware
- Compression and Your own middleware

Hands-on

- Handling form-URL-encoded requests in a simple signup mechanism
- Using sessions to persist data
- Write a simple route protection middleware

Routing

Learning Objectives

Learn to implement modular routing using the Express Router module. Organize the routes by service domains and learn to handle dynamic routes and route parameters. Finally, implement a very basic route protection mechanism using middleware functions.

Topics

- Modular routing with Express Router
- Organizing routes
- Dynamic Routing and Route parameters
- Basic route protection using middleware

Hands-on

- Build a simple API
- Use route parameters to render a list of movies and genres

Database Integration

Learning Objectives

Set up a database on MongoDB Atlas and configure it to work with the Express application. Learn to set up and configure Mongoose, the most popular ODM for MongoDB. Next, build a Mongoose Schema and Model to manage user accounts/store and manage blog posts. Finally, bring in our precooked React web application and set it up.

Topics

- Working with MongoDB Atlas
- Setting up Mongoose ODM
- Building the Users schema and model
- Building the Posts schema and model
- Serving the client web application

Hands-on

- Setting up Mongoose ODM
- Building a Schema and Model
- Integrating a client application

Authentication, Controllers, and APIs

Learning Objectives

Understand all about JWT Authentication and how it differs from session-based authentication. Build a controller that enables admins to login. Also, build controllers and API for letting users signup or log in from the client app. Understand how to build controllers and API for creating, reading, and deleting blog posts. Implement profanity filtering and the ability to moderate posts.

Topics

- Signing up an Admin and Password Hashing
- Admin Authentication - Controller and Route
- Authentication API - Controller and Routes
- Posts API - Controller and Routes
- Profanity Filtering and Post Moderation

Hands-on

- Creating a controller

Caching and Performance

Learning Objectives

Level up the caching strategy by setting up Redis and implement Redis as a cache. Understand how to use Redis as a fast session store.

Topics

- Setting up Redis for caching
- Caching and Serving Content
- Configuring Redis as the session store

Protecting Express apps

Learning Objectives

Learn to handle untrusted data such as form input. Also, learn to mitigate XSS and CSRF attacks.

Topics

- Handling untrusted data
- Preventing XSS and CSRF

Hands-on

- Handling untrusted data

Deployment

Learning Objectives

Learn to deploy the Node and Express app on Heroku.

Topics

- Deploying an Express app on Heroku

Capstone Project

For the final capstone project, you are required to build an API server that uses a file-based database (or MongoDB) to serve multiple routes and HTTP verbs with minimal configuration needed to get up and running. The result is a simple, yet powerful API server for development purposes such as building React, Angular, and Vue applications.