



NodeJSSyllabus

Course Objectives:

- Node.js uses JavaScript - a language known to millions of developers worldwide - thus giving it a much lower learning curve even for beginners.
- Using Node.js you can build simple command line programs or complex enterprise level web applications with equal ease.
- Node.js is an event-driven, server-side, asynchronous runtime environment.
- Node.js helps you to code the most complex functionalities in just few lines of code.

Getting started with Node.js

- Hello World HTTP server
- Hello World command line

- Hello World with Express
- Installing and Running Node.js
- Debugging Your NodeJS Application
- Hello World basic routing
- Hello World in the REPL
- Deploying your application online
- Core modules
- TLS Socket: server and client
- How to get a basic HTTPS web server up and running!

Npm

- Installing packages
- Uninstalling packages
- Setting up a package configuration
- Running scripts
- Basic semantic versioning
- Publishing a package
- Removing extraneous packages
- Listing currently installed packages

Node.js Event loop

- The Node.js process

File System

- The FS Module.
- Reading Directories
- Reading Files
- Streams
- Building servers
- Creating servers with HTTP
- The Http server class

- Parsing requests
- HTTP streaming
- Building APIs using modules, events and packages
- CommonJS Modules
- The global object
- Directories as Modules
- npm Packages
- Managing Dependencies
- npm scripts
- Async Patterns
- Asynchronous Javascript
- Asynchronous Control Flow with Callbacks
- Promises
- Async/Await
- The EventEmitter API
- Third Party Async packages
- ExpressJS
- REST
- Introduction to ExpressJS
- Routing
- Responding
- Configuration
- Views
- Middlewares
- Receiving Data
- Error Handling
- Working With Data
- Authentication With Passport and JWT

- Stateful vs. Stateless Authentication
- OAuth2
- Passport
- JWT – JSON Web Tokens
- Hosting Standalone Node
- Why do we need hosts
- Forever
- PM2
- Node on Windows
- Node as a Windows Service
- IISNode
- Hosting Complex Node Architectures with Docker
- What is Docker
- Docker CLI
- Docker File system & Volumes
- Docker Files
- Containers Communication
- Docker Compose
- Best Practices
- NPM
- Best Practices
- Node.js API design
- Error Handling
- Debugging