



# Node.js & its Security Frameworks

---

---

SIT209-Topic9

Above and Beyond-Project Report

Vansh Khanna, 2010993022

Vansh3022.be20@chitkara.edu.in

## Overview

Node.js is a network application development platform. It's an open-source server-side environment that helps us in building scalable network applications. It uses JavaScript to write code that can be run directly in the Node.js runtime. It's compatible with Microsoft Windows, Linux, and Mac OS X. It also includes a collection of libraries containing various JavaScript modules that may be used to construct web apps.

## What can we do with Node.js?

With Node.js, one can develop JavaScripts that run outside of the browser on a cross-platform, open-source, back-end JavaScript runtime environment. JavaScript can also be used to write command-line tools and for server-side scripting, which generates dynamic web page content server-side before a web page is sent to the user. Hence, Node.js represents a "JavaScript everywhere" paradigm, thus enabling web application development around a single coding language, rather than using different scripting languages for server-side and client-side operations.



## Why Node is Special?

- Prototyping and agile development are made easy
- Creating services is lightning quick and scalable
- Supports the frequently used language JavaScript
- The codebase is cleaner and more consistent.
- A large open-source library ecosystem

## What is a Node framework?

The node framework consists of libraries and tools which are required to create a software application. It provides a basis on which different software applications can be built. The node framework is a workspace platform that supports Node.js and lets developers build



front-ends and back-ends of applications using JavaScript. There are a variety of Node frameworks that extend the features and functionalities of Node.

## Types of Node.js Frameworks

Here is a tiny list of framework types that have been recognized by Node.js, along with the numerous libraries to be utilized in diverse scenarios such as:

### I. MVC Frameworks

#### A. Sinatra-Like:

This framework has more configuration options than Rails-like or full-stack frameworks and is less opinionated.

#### B. Rails-Like:

Helps in presenting projects properly. One can create their own art forms from the included templates and libraries.

### II. Full-Stack Frameworks

Scaffolding, template engines, WebSocket, and persistence libraries are all included in full-stack MVC frameworks, allowing one to create real-time, scalable web projects. So here Node.js framework shines the brightest.

### III. Rest API Frameworks

It is made for those users that require a quick Node.js REST API server and don't use rich-client/front-end MVC frameworks.

### IV. Other Libraries

It includes different middleware, libraries, and static site generators.

## Popular Frameworks

Let's take a look at some of the below listed most popular frameworks on the market these days:

## 1. Express.js



Express.js is a flexible and lightweight Node.js application framework created by TJ Holowaychuk that is specifically built for constructing single-page, multi-page, and hybrid apps with a rich set of functionality for web and mobile applications.

There is no built-in object-relational mapping engine in Express. Express has "no opinion" on what technologies you plug into it because it isn't developed around certain components. Express is an excellent contender for agile development and rapid prototyping because of its independence, as well as its lightning-fast setup and Node's pure JavaScript environment. Startups who want to build a product as rapidly as possible usually don't have a lot of legacy code use Express.

All of the core components of the framework are updated and reformatted on a regular basis.

### Features

- It has a quick, dependable, and asynchronous framework
- It aids with server and route direction.
- Assists in the creation of a variety of web applications.
- It's known for its high-speed I/O, nonparallel architecture, and single-threaded nature.

## 2. Hapi.js



It is introduced by Eran Hammer at Walmart while trying to handle traffic on black Friday. It is an open-source, stable, and consistent Model-View-Controller MVC framework for constructing web applications and services which allows one to

create API (application programming interface) servers, websites, and HTTP (Hypertext Transfer Protocol) proxy applications with ease.

It allows one to quickly add new features and solve faults due to the plugin's system robustness. Routing, input, output validation, and caching are all elements of Hapi.js that helps in structuring REST APIs.

The well-developed plugin architecture and several critical features like input validation, implement caching, configuration-based functionality, error handling, logging, and more make the Hapi one of the most favored frameworks. Several large-scale websites, such as PayPal and Disney, use it to construct helpful applications and provide technology solutions.

## Features

- Code reusability
- No external dependencies
- Security
- Integrated Architecture: a node framework with a complete permission and authentication API.

## 3. LoopBack.js



LoopBack is a Node.js framework with a dynamic API explorer and an easy-to-use CLI that allows developers to easily set up models and build sophisticated end-to-end REST APIs with minimal coding. It aids with the model relation management, add-on segments, multiple backend data stores, and ad-hoc searches, as well as secure login and authorization settings. It also supports a variety of REST services and databases, including MySQL, Oracle, MongoDB, Postgres, and others.

Using this method, one can create a server API that maps to another server, thereby acting as a proxy for another API. If one does not have a schema, it is also possible to generate dynamic models and models based on one's schema. Furthermore, it provides native mobile and browser SDK compatibility for clients like Android/Java, iOS, and Browser JavaScript (Angular).

## Features

- Unbelievably extensive
- Graph QL support

## 4. Nest.js



NestJS is a Node.js framework that helps developers create modular, scalable, and maintainable server-side web applications in every way. It follows the MVC (Model-View-Controller) design paradigm and is extensible. NestJS's native support for TypeScript is a standout feature, providing access to optional static type-checking, powerful tooling for huge apps, and the most up-to-date ECMAScript capabilities. It also supports TypeScript, as well as some aspects of Functional Programming (FP), Object-Oriented Programming (OOP), and Function Reactive Programming (FRP) (FRP).

### Features

- Extensible: Its modular architecture allows it to use any other libraries, making it really adaptable.
- Versatile: Its adaptive ecosystem serves as a full-fledged backbone for a wide range of server-side applications.
- Progressive: Takes advantage of the latest JavaScript features to bring design patterns and smart solutions to the world of node.js.

## 5. Adonis.js



Adonis is a node.js framework with a hardcore MVC structure that focuses on being a reliable framework in the Node framework colony, it includes a variety of test modules that aid in the optimization of one's code to make it more stable. Furthermore, its design pattern divides distinct functionalities into different portions of an application and employs the edge template engine, which is extremely user-friendly.

### Features

- It has its own CLI (Command Line Interface).

- Similar to Laravel, making it simple to learn.
- Validators examine if the data going into the controllers is in the correct format and send out messages if there are any issues.

## Conclusion

Discovering new Node JS frameworks can be overwhelming, and it's likely that you'll need to do a lot of research before getting started. The Node JS frameworks mentioned above are the most widely used and provide a variety of capabilities.

## References

1. "Node.js - Wikipedia," En, <https://en.wikipedia.org/wiki/Node.js>.
2. S. Tilkov and S. Vinoski, "Node.js: Using JavaScript to Build High-Performance Network Programs," IEEE Internet Computing, vol. undefined, no. undefined, pp. Page(s): 80 - 83, Dec. 2010, doi: undefined.
3. "Node.js Frameworks," Nodeframework, <http://nodeframework.com/>.
4. "Express - Node.js web application framework," Expressjs, <http://expressjs.com/>.
5. "hapi.dev - The simple, secure framework developers trust," Hapi, <https://hapi.dev/>.
6. "LoopBack," Loopback, <https://loopback.io/>.
7. "NestJS - A progressive Node.js framework," Nestjs, <https://nestjs.com/>.
8. "AdonisJS - A fully featured web framework for Node.js," Adonisjs, <https://adonisjs.com/>.