**Node.js frameworks:**

**Express.js**

This is the most well-known framework used by Node.js developers. Minimalist and with a fairly simple learning curve, it has a very complete ecosystem to execute what you need, with the functionalities separated into different modules.

It is used, among other uses, for:

* Developing REST APIs that are used to communicate with client applications.
* Creating traditional MVC applications.

Features:

* Speedy server-side development
* Amazing Performance
* Extensive options in tools
* High test coverage
* Content Negotiation
* Efficient Routing
* Generates application seamlessly
* Easy debugging
* Precise responses to HTTP requests by enabling the setting up of middleware

Express.js is powered by very high-end features like detailed routing, configuration, security features, debugging mechanisms, etc. Therefore Express.js framework is well-suited for the web apps of enterprise-level.

**Hapi.js**

Similar to Express.js, it was born at Walmart to handle traffic at events like the famous Black Friday and its main objective is to support numerous requests. Its fame is due to its high performance. The developer community believes that Hapi is robust and reliable, because it was able to successfully manage millions of shopping transactions at Walmart during Black Friday.

Solid and complete, with many libraries available, it is used for:

* Same projects as Express.js, but is more focused on getting developers to focus on writing application logic.
* High performance applications.

**Features:**

* js is extremely secure as every line of code is thoroughly verified
* The presence of multiple official plugins makes it a rich ecosystem.
* Easy and straight implementation
* Predictability
* Simple interface

Hapi.js is empowered by superior functionalities like body parsing, input/output validation, HTTP-friendly error, etc. It is also secure and reliable. If you are someone that wishes to implement Express for your web app, we suggest you go with Hapi.js.

So, these were some of our most loved and best Nodejs frameworks for app development in 2023.

**Nest.js**

This is a more modern framework that uses Typescript and is influenced by Angular. Designed for current development and optimized to work with architectures and clean code, it is constituted in a decoupled way, by modules. It also has an important community that is growing by leaps and bounds.

It is useful for:

* Performing REST APIs services.
* Creating MVC services.
* Working with hexagonal architecture or other complex architecture.

**The Pros**

Some of the benefits of using Nest.js include:

Powerful but user-friendly: The framework is developer-friendly to work with, enough that even the most complex and powerful features are easy to use. The Nest.js team designed the framework for developers to quickly get started and focus solely on writing the business logic while the framework takes care of other important aspects of development, [such as security](https://kinsta.com/blog/website-security-check/).

Angular-style syntax (backend): [Angular is a very popular frontend framework](https://kinsta.com/blog/angular-vs-react/) with a concentration on architecture and structuring. Nest.js acts as Angular for the backend because it uses Angular style and syntax to help you structure your enterprise project.

TypeScript: Nest.js supports TypeScript right out of the box, and this solves performance and writing maintainable applications quickly by providing compile errors and warnings. TypeScript makes it well-integrated into VSCode for an accessible development environment.

Thorough documentation: Nest.js features some of the best documentation for any framework, which is very easy to understand. It saves debugging time to quickly glance through the documentation and be greeted with the solution to your problem.

Good architecture and fast development: Nest.js saves you time when building out your application, whether you’re creating your first MVP or the actual application, because it gives you a solid structure and architecture to work with out of the gate, thereby enhancing your development processes.

**The Cons**

* + There is a famous saying: “Every advantage has its disadvantages” — and Nest.js is no exception. So, let’s explore the cons of Nest.js.
  + Difficult for novices: Nest.js can be a bit tricky for novice developers who are not coming from an Angular background to learn and master. Additionally, since not every JavaScript developer uses TypeScript, the framework can be difficult for these sets of developers too. But just like every other technology, it requires time and practice.
  + Arduous debugging: As much as TypeScript has its gains and benefits, it also comes with many debugging pains, especially for newer developers in the TypeScript world.
  + There you have it: the pros and cons of Nest.js and how they can influence your decision-making in choosing it as your framework of choice.

**AdonisJS**

Ideal for developers linked to Django, Rails, Laravel, etc., it provides an important variety of functions so that you can concentrate only on developing the project and save time by avoiding the installation of specific libraries.

Designed for:

* Developing more complex projects that need numerous features, since it comes with “batteries included”.

**The Pros**

* Solid project structure
* AdonisJS provides easy yet powerful validation rules

**The Cons**

* Small community with limited resources