

FAST-TRACK WEB APPLICATION DEVELOPMENT WITH NODE.JS



CONTENT



1. Introduction
2. What Node.js is?
3. Reasons to use Node.js
4. Application of Node.js
5. Conclusion

Introduction

Ever-changing technology has become an inseparable part of today's life and so of global organizations. Organizations are launching new products almost everyday, that aim to resolve the roughest business challenges, help them to attract potential customers and eventually outperform competitors.

Innovation or disruption isn't enough for an organization to win. It is also about how quickly, swiftly and error-free they innovate and bring products to market. In this age of agile development cycles, work doesn't end at delivering an application or functionality. In fact, it is more important to understand the user needs and aim to improve user experience through these functionalities and update the application as per the newest technologies.

Recently, Node.js is gaining adhesion among developers and organizations for building scalable and high performing web applications, IoT and many other products.

According to one of the recent surveys by The Node.js Foundation,

“98% of all the organizations use Node.js regularly”



What Node.js is?

Built on Chrome's V8 JavaScript engine, Node.js is an open-source, cross-platform runtime environment for developing server-side web applications. It executes JavaScript code outside of a browser. Node.js allows developers to use JavaScript for writing command-line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Node.js represents a "JavaScript everywhere" paradigm, unifying web application development around a single programming language, rather than different languages for server- and client-side scripts.

Node.js allows developers to create Web servers and networking tools using JavaScript and a collection of modules that handle various core functionalities. Modules are provided for file system I/O, networking, binary data, cryptography functions, data streams, and other core functions. Node.js's modules use an API designed to lessen the complexity and pain of writing server applications.

The difference between Node.js and PHP is that most functions of PHP are of **blocking** nature until completion (commands only execute after previous commands finish), while Node.js functions are **non-blocking** (commands execute concurrently or even in parallel, and use callbacks to signal completion or failure). Developers and coders across the globe are using Node.js to build high performing and scalable applications including enterprise applications, API engines and mobile websites.

Reasons to Use Node.js

1. JavaScript – A Dynamic Language

JavaScript is one of the most commonly used programming languages used in web browsers which allows client-side scripts to communicate with the user. Technically speaking, it joins front-end and back-end development. It is an object-oriented and event-based language which has the in-built capacity to handle the event loop mechanism in a simpler way. Node.js has broader footprints as JavaScript runs across various OS environments that support them in any browser.

2. Asynchronous Programming

Node.js uses JavaScript which is asynchronous in nature. Asynchronous refers to the occurrence of events independent of the main program flow and ways to deal with such events. It is a programming method which provides a feature of non-blocking code and allows Node.js to callback events to a single thread.

Async is parallel programming in which a unit of work runs separately from the main application thread and notifies the calling thread of its completion, failure or progress. Asynchronous programming improves performance, system throughput, resource utilization and works faster. Node.js has numerous modules that control the flow of a program and one of the most common modules is Async.

3. Community Driven

Node.js has a flourishing open source community of members readily available to help each other. Members of this community have produced multiple modules and a shared library of codes that can be freely accessed through Node Package Manager (NPM). NPM runs through the command line and packages can be installed locally as well as globally.

4. Node Package Manager – NPM

The Node Package Manager (NPM) is an online repository of Node.js projects for developers. There are thousands of modules and packages freely available to the Node.js community. The complete list of modules and reusable components can be accessed using NPM CLI tool that automatically gets installed with Node.js. Some of the common and most popular NPM modules are – Express.js, Connect, Socket.io, Mongo.js, Redis, CoffeeScript.

5. Handles Concurrent Requests

One of the biggest advantages of using Node.js is its ability to handle concurrent connections with high throughput. The backbone of a Node.js application is the single-threaded event loop, which repeatedly picks up an event and provides it to any of the event handlers, one at a time. As the event handlers wait for more events, the other computations are carried forward concurrently, allowing a Node.js process to handle the massive amount of requests.

6. Fast Paced

Loaded with V8 engines powered by Google, Node.js compiles JavaScript into a native machine code running pretty fast. This is why it can speed up any framework.

7. Creating a real-time web application

Node.js is a boon for those looking to simplify their tasks for real-time application development. For those web applications that require using an event-based and non-blocking server, the real-time web application Node.js fulfils all the requirements.

8. Single codebase

Sending data between the server and client for effective synchronization is not an easy task. But using Node.js and writing the code into JavaScript for server and client both and the data synchronized automatically. Node.js has the ability to execute a similar code on both the sides – client and server. Meteor a web app framework built on top of Node.js executes the same codebase on both the client and the server. The data changes on the server immediately reflect in the client and the webpage that displays this data automatically updates itself.

9. Data streaming

Any successful product development would require top-notch I/O handling. With Web frameworks treating HTTP requests and responses as whole data objects this gets tough. Node.js is the solution for this since it's good at handling such I/O process which allows users to transcode media files simultaneously while they are being uploaded. Reading and writing stream to Web-Sockets enables handling such HTTP in a rather easier manner,

10. Simplified hosting

As more companies prefer Node.js, the platform is adding more facilities and multiplying the pace of introducing new features. This offers great help to those involved in providing Platform-as-a-Service (PaaS) services that reduce organizations to a solitary order. Perhaps, this is the reason why even Heroku prefers Node.js.

11. Being SEO friendly

Top ranking while Search Engine Optimization (SEO) is something everyone wants. The backend rendering by Node.js gives the website more visibility as search engines can optimize the React projects. The high speed and top-notch performance also contribute to bettering the SEO prospects along with quality user experience.

12. Using open toolset

Open Source development companies are using Node.js since it offers a very engaging development environment for free. This includes all the necessary tools for developers like Redux and Flux while running Node.js on the backend. Plenty of Node.js development services are making the most out of open source libraries for task-specific activities.

13. Deploying of apps get easy

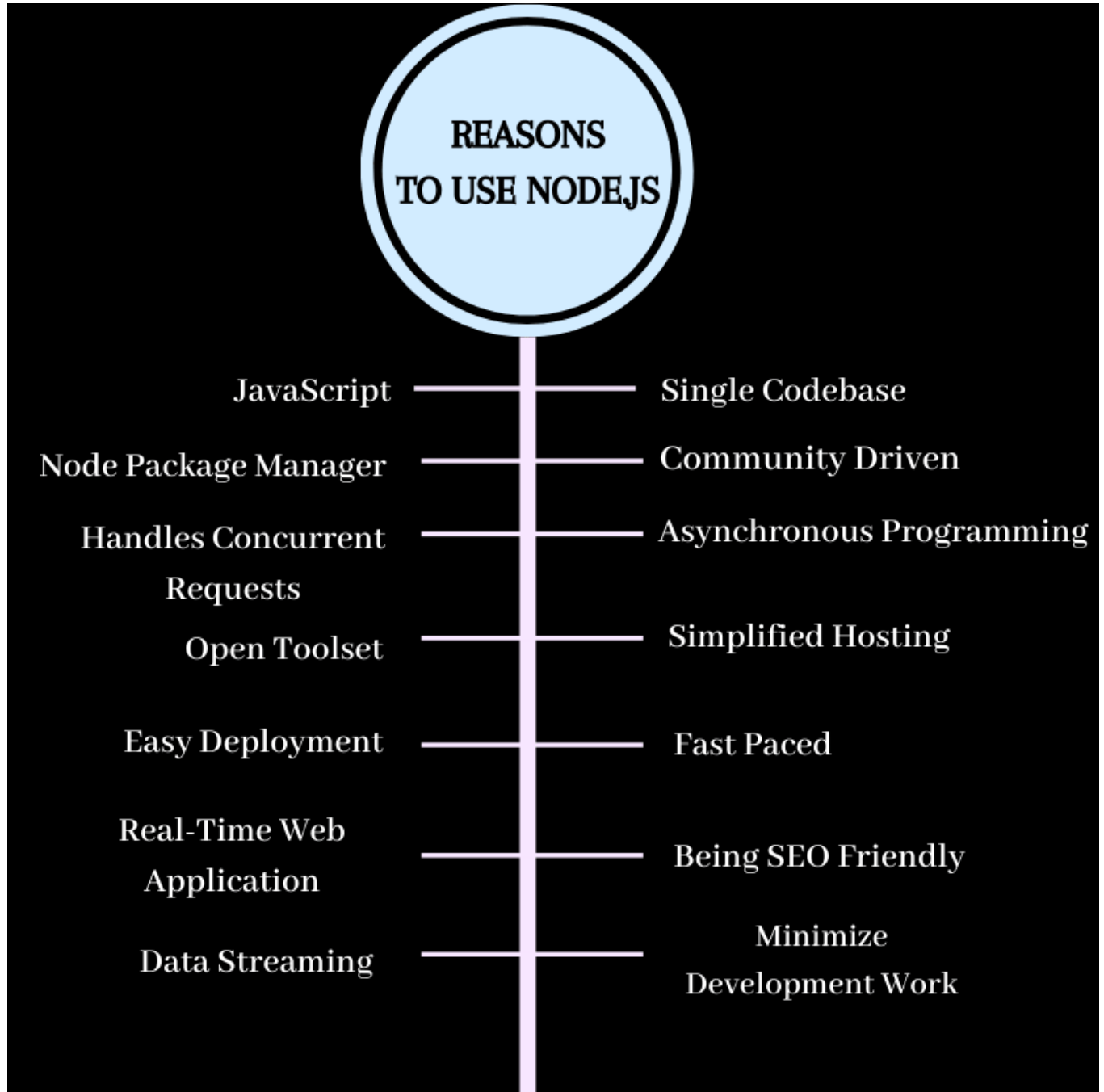
Node.js offers comparable solutions for projects requiring a dedicated app for programming environments. Due to its dynamic ecosystem and dedicated setup, project completion is simplified considerably.

14. Minimize development work

Node.js has a range of resourceful frameworks and libraries available so it makes working on a project a lot easier. During the development of web applications, developers create a pool of such resources for multiple usages. This helps to reduce the development time by writing reusable utility classes instead of starting it from scratch every time for a new project. Instead, organizations using Node.js can capitalize on this to have Node.js developers write business-specific logic to gain further sales from it. With the increased adoption of Node.js, hosting is also growing. PaaS (Platform as a Service) service providers such as Heroku and Modulus are now formally supporting node deployments to a single command.

An organization that uses Node.js can-

- Use ½ the number of programmers to build a new product.
- Make Developers and Customers Happy
- Reduce the page load times by 50%
- Retain employees for longer
- Reduce the number of servers needed to host the application



Application of Node.js



Chat

Chat is the most typical real-time, multi-user application. A chat solution can be developed using Node.js as it requires high traffic but low processing power.

The chat application is really the sweet-spot example for Node.js: it's a lightweight, high traffic, data-intensive (but low processing/computation) application that runs across distributed devices. These days it is true that most people use phone-based apps or social media platforms to chat with friends. Uses of Node.js can create chat rooms very easily. Independent chat rooms are probably not very popular on today's Internet, but this is used a lot in online games which makes more fun when there is a chat component to them.



Data Streaming

With the concept of a callback, HTTP requests and response in Node.js streaming data flow can be easy. It enhances real-time audio and video encoding and boosts the performance of the application.

In more traditional web platforms, HTTP requests and responses are treated like isolated events; in fact, they're actually streams. This observation can be utilized in Node.js to build some cool features. For example, it's possible to process files while they're still being uploaded, as the data comes in through a stream and we can process it in an online fashion. This could be done for real-time audio or video encoding and proxying between different data sources.



Payment Processing

Node.js is one of the top platforms that makes payment or credit card processor API integration simple & robust on the server-side.



Stock Trading Software

Node.js is an ideal platform to build stock trading applications that track stock prices in real-time, perform calculations and show detailed analysis.



Real-Time Monitoring Applications/Tools

Node.js is best-suited technology for organizations to develop an application or tool that tracks website visitors and their interaction in real-time. Node.js lets developers monitor performance metrics from the application to the operating system in real-time, including response time, CPU utilization, disk I/O, slow web service calls, slow SQL calls, memory and more.

One can -

- Track application response time
- Monitor dependencies and API configurations
- Pinpoint event loop lags
- Check CPU usage
- Plug Memory leaks



The IoT

IoT (Internet of Things) is a network of devices such as sensors, beacons, actuators, and any other items embedded with electronics that enables them to send and exchange data. Normally,

IoT systems pass data from devices to servers and from servers to applications that process it and display it to users. IoT may consist of thousands of such devices, which makes it challenging to manage requests and data streams from and between the devices. Node.js is the best suited for IoT as it requires high interactivity and secure communication with APIs connected through various devices and applications.

Node.js is one of the preferred solutions for enterprises and organizations to develop their private and public IoT systems. The most obvious advantage of Node.js as a back-end for such networks is its ability to process multiple concurrent requests and events emitted by thousands or even millions of devices on the network. The avalanche of requests and data coming from IoT devices does not block Node.js servers. This makes Node.js fast as an application layer between these devices and databases used to store data originating from them.



eCommerce

Node.js is a powerful platform for developing e-commerce applications that aim to provide payment gateway integration, reporting and analytics, third party application integration, vertical and horizontal stability etc. Node.js as part of JavaScript tech stack was a good fit for this e-commerce giant. Node.js has become their front-end web stack and it powers many of eCommerce web flows.



Social Media

LinkedIn is the best example of a social networking site built on Node.js. The amount of work to be done on the server-side can be greatly reduced and made viable for large scale deployments. Performance efficiency and scale were two primary reasons for the social medias to switch to Node.js.



Enterprise Web Services

Node.js contains applications in smaller modules that are connected together. This makes it possible to build smaller applications and integrate them with the larger ones. This makes Node.js powerful, flexible and more used. According to the Node.js User Survey, 43% of Node.js programmers claim to have used it for enterprise apps - and, interestingly, the majority of them seems to be working for Fortune 500 companies.

Conclusion

Node.js is not just a web-server anymore it is also the go-to tool for developers using unit testing and static code analysis tools that run on Node.js. Node.js is becoming a technology of choice with a huge community base working on the ecosystem every day.

Node.js is great for building applications with heavy client-side rendering, multiple concurrent requests, and frequent shuffling of data from a client to a server. Node.js an emerging technology that provides lots of vital features like garbage collection & process memory, event looping:

Node.js's ability to be CPU bound and use async operations, so CPU can be highly utilized and no wastage of waiting for cycle for I/O Operation. Although, Node.js is an asynchronous, non-blocking event-driven approach which allows a developer to build a lightweight real-time scenario application with ease of use such as Messaging apps, Chat Apps, and so on. It empowers developers as well the users to utilize the web application in its full capacity and yield desirable results.