**Top Five Reasons to use Node.js with React.js for Web application –**

[Krina Sardhara](https://medium.com/@krina_sardhara?source=post_page-----ce96e759c581--------------------------------) Jun 15, 2022

While building an app, the first question that usually arises is about choosing a technology stack. The technology we use has the highest impact on app performance and quality. In this blog, I talked about one of the most popular stacks that I personally used in my web app.

**ReactJS** and **NodeJS**have been two of the most popular topics in the developer community in recent years. Both technologies have their unique way to improve development ***efficiency***, ***quality***, and ***scalability***.

***Node.js is a back-end Javascript runtime environment*** built on Chrome’s V8 Javascript engine. It uses an event-driven, non-blocking I/O model and is built for scalability and high performance.

On the other hand, ***React.js is the free, open-source, front-end JavaScript Library*** that runs on the browser and is used to create user interfaces. It is developed and maintained by Facebook.

Now the question

**Could we use React with NodeJS for Web application development?**

**Yes**, without a doubt, NodeJS is the most convenient platform for developing reliable web servers for React applications.

Node js works alongside the Node Package Manager (NPM) which consists of an online repository for publishing open-source Node.js projects, which can be easily installed through a command-line interface.

Node bundles React app into a single file for easy compilation using node modules such as Webpack. The V8 engine in NodeJS can handle bulk requests via callback functions (V8 is the JavaScript engine inside of node.js that parses and runs your JavaScript).

NodeJS and ReactJS both use **JavaScript** language that can be executed on the server and client sides, and therefore React.js code can be directly executed in the Node.js environment.

Further, the React DOM components are specifically designed to work with NodeJS, reducing the number of lines of code and simplifying server-side rendering.‍

**Is it compulsory to use React and NodeJS together in web application development?**

Some may believe that Node.js is required as a backend to run React.js. However, using Node as a backend for React is not compulsory; you can use another backend technology such as Laravel, Ruby, and Django to deal with your backend processing.

Node is highly efficient in handling real-time data processing in the backend and is mostly used in applications where you need to handle a large number of simultaneous network connections in a non-blocking manner.

While React is a library that only deals with building UI components, it is used to render the user interface of your mobile and web applications. It uses different tools like React-router (Uses component structure to call UI components) and Redux (Manage application state) to handle server-side communication by Node.

As a result, which technology to use is determined by project requirements and developer convenience. Before concluding the project technology stack, developers must study and analyze their use cases.

If your project has specific conditions and requirements that can be effectively managed or fulfilled using NodeJS for React, you must go for it.‍

**Top 5 reasons to use React with Node.js**

Here are 5 reasons to use React and NodeJS to simplify your web development and make your code highly efficient and scalable.

**1. Helps to maintain a high server load**

Having Node in your React project makes it super easy for servers to handle multiple network requests and manage server load balance.

**2. Helps to maintain real-time data**

If your app demands real-time data management or data streaming, the combination of React and NodeJS is extremely beneficial. This is because the Node is best for managing continuous server connections and React for front-end data rendering.

**3. Helps in developing single-page applications**

Using React and Node js, you can easily create single-page applications like Gmail and Facebook. React allows you to create intuitive UI/UX while Node provides asynchronous data loading via callback.

**4. Simplify building JSON for your APIs**

JSON stands for JavaScript Object notation- An internet data serialization format that helps in the transmission of data between website and server.

JSON API is great for making JSON response formatting more consistent, and its efficient caching features can eliminate superfluous server requests. Using JSON API with Node is very efficient, it promotes code reuse. The reusable code gets sharable using React.

**5. Technologies of the MERN stack project**

[MERN Stack application](https://www.dhiwise.com/post/create-mern-stack-applications) uses mainly four technologies- MongoDB, Express.js, React.js, and Node.js. With this stack, we can create applications like e-commerce and banking websites. Along with MongoDB, Express.js, Node.js, and React.js, developers only need to know JavaScript and JSON to build apps with ease.

You now have a solid understanding of Node.js and React.js, as well as when to use them.

If your web project has similar requirements to the ones we discussed above then you can take the advantage of both the technologies React and Node js to improve team productivity and efficiency.

Furthermore, if you want to accelerate your application development try DhiWise which supports NodeJS, ReactJS, Express.js, MongoDB, and other popular technologies.

[***DhiWise***](https://www.dhiwise.com/?utm_campaign=blog&utm_source=seo&utm_medium=medium&utm_term=education&utm_content=node)***is a 100% developer-centric platform that aims to speed up app development.***It simplifies web and mobile app development with its ultra-modern features such as,

* Support for multiple technologies and frameworks: [Node js](https://www.dhiwise.com/node?utm_campaign=blog&utm_source=seo&utm_medium=medium&utm_term=education&utm_content=node), [React js](https://www.dhiwise.com/react?utm_campaign=blog&utm_source=seo&utm_medium=medium&utm_term=education&utm_content=node), [Laravel](https://www.dhiwise.com/laravel?utm_campaign=blog&utm_source=seo&utm_medium=medium&utm_term=education&utm_content=node), Express js, MongoDB, [Android](https://www.dhiwise.com/android?utm_campaign=blog&utm_source=seo&utm_medium=medium&utm_term=education&utm_content=node), [iOS](https://www.dhiwise.com/ios?utm_campaign=blog&utm_source=seo&utm_medium=medium&utm_term=education&utm_content=node), and [Flutter](https://www.dhiwise.com/flutter?utm_campaign=blog&utm_source=seo&utm_medium=medium&utm_term=education&utm_content=node)
* Design to code conversion in minutes: Figma, Adobe XD and Sketch to React, Flutter, Android(Kotlin), and iOS(Swift)
* Flexibility to create and change app-specific workflow with the Logic builder: Configure Model, set up permissions, Payment APIs, Authentication, routes, and more.
* Conditional flow for mobile apps: Set up conditional flow for mobile applications.
* Integration with GitHub and GitLab to simplify code sharing among teams.
* Choice of architecture selection: Select MVC or MVVM/Clean code architecture according to your preferences.
* Connect with the database schema of your choice with PostGRES, MySQL, MongoDB, and standard SQL server.

Want to know more about **[DhiWise](https://app.dhiwise.com/sign-up?utm_campaign=blog&utm_source=medium&utm_medium=cms&utm_term=education&utm_content=node" \t "_blank)** and its capabilities? [**Sign up**](https://app.dhiwise.com/sign-up?utm_campaign=blog&utm_source=medium&utm_medium=cms&utm_term=education&utm_content=node) on DhiWise for free & Start using **[DhiWise](https://app.dhiwise.com/sign-up?utm_campaign=blog&utm_source=medium&utm_medium=cms&utm_term=education&utm_content=node" \t "_blank)**to experience the ease of app development while having full control over your source code.