**Node.js vs React.js: which to choose for your web app project**

Harikrishna Kundariya Sep 10, 2020

With JavaScript being one of the most popular languages out there with so many JavaScript frameworks to choose from. Node.js and React.js are two of the most popular options out there, but developers sometime struggle to know which one is better.

The main difference is that Node.js is a back-end framework, and React.js is used for developing user interfaces. Both frameworks are quite popular and have their advantages and disadvantages.

In this article, we compare Node.js and React.js based on their performance, knowledge curve, community, and microservices support.

**Today, we will cover:**

* [React Overview](https://www.educative.io/blog/nodejs-vs-reactjs?eid=5082902844932096#react)
* [Node.js Overview](https://www.educative.io/blog/nodejs-vs-reactjs?eid=5082902844932096#node)
* [Which is better?](https://www.educative.io/blog/nodejs-vs-reactjs?eid=5082902844932096#better)
* [What to learn next?](https://www.educative.io/blog/nodejs-vs-reactjs?eid=5082902844932096#next)

**React Overview**

React.js is an open-source JavaScript library. This front-end framework allows you to develop user interfaces for websites and apps in an organized way. React is loved for its high performance.

React was developed by Facebook for everything from content-based websites to mobile apps. Both Facebook and Instagram apps were developed with React. The library of React.js also helps Netflix improve runtime performance, modularity, and rapidity.

**Features of React.js**

* **Simple support:** This library re-utilizes framework parts and resolves any issues associated with React refreshes.
* **Execution:** React utilizes the Virtual DOM (document object model), which improves application performance.
* **Stable code:** By utilizing descending information streams, React.js ensures your code’s soundness and execution.
* **Designer toolset:** React.js furnishes developers with high-performance design tools, guaranteeing better performance.
* **App development:** By following similar plan designs, you can construct rich UI for native applications upheld by iOS and Android.

**Advantages of React.js**

**Better Efficiency:**

React allows you to keep all app features in its own virtual DOM. All the changes done in DOM are automatically updated.

**Web Optimization Friendly:**

Using React JS, you can make lightweight web applications. One can run React on the server-side, and the virtual DOM will be delivered.

**Component-based Construction:**

You can easily make segments per your requirements. It allows you to create your components and integrate them into your core content. React.js updates rapidly and ships well with smooth interface designs.

**Limitation of React.js**

**Difficult to learn:**

The framework is quite complex and it requires in-depth knowledge for integration into a user interface.

**Poor documentation:**

React technologies have frequent updates, and it accelerates rapidly. Due to this, some complain that the documentation is hard to use.

**View part:**

React.js framework covers only the UI layers of the app. So, developers still need to use some other technologies to get a complete tool set.

**Node.js Overview**

In basic terms, Node is an open source cross-platform library for server-side programming that permits clients to develop web applications rapidly. With Node, we can execute JavaScript applications or network applications. Its basic modules are engraved in JavaScript.

It is generally utilized for server applications in real-time. Node.js permits JavaScript to execute locally on a machine or a server.

Node.js gives numerous systems to utilize. One of such structures is Express.js. It is more valuable and mainstream than the different structures of Node.js.

**Features of Node.js**

* **Versatility:** Node is incredibly adaptable as the server reacts in a non-blocking way.
* **Zero Buffering:** Applications yield the measurements in enormous pieces. This gives the advantage of ‘No buffering’ to developers.
* **Network:** Node.js upholds an open-source community. This is the main explanation that numerous glorious modules have been added to Node.js applications over time.
* **Occasion driven Input and output:** APIs of Node.js are non-blocking, meaning that the server won’t wait for the arrival of information from an API. Rather, it will move to another API.

**Advantages of Node.js**

**Easy to learn:**

Nodeis quite simple for developers to utilize and learn. Learning Node.js is less difficult than React.

**Better Performance:**

Node.js takes the code of JavaScript via Google’s V8 JavaScript engine. The main advantage of this process is that it complies with the JavaScript code directly into the machine code

**Freedom:**

Node.js offers a lot of freedom when it comes to development. There are generally less constraints with Node.

**Extended support for tools:**

Another advantage of Node.js is that developers have more community support.

**Extensible:**

Node.js is known to be quite extensible. You can utilize JSON to give the degree to trade of information between the web server and the client.

**Scalability:**

Node.js makes it simple to scale applications in horizontal as well as vertical directions. The applications can be scaled even by the option of extra hubs to the current framework.

**Limitations of Node.js**

**Programming interface isn’t steady:**

The Application Programming Interface (API) of Node can be challenging to work with. It changes regularly and doesn’t remain stable.

**No strong library support system:**

JavaScript does not hold a strong library system. This limits the developers to implement even common programming tasks using Node.js.

**Programming model is not synchronous:** Many developers find this programming model tougher in comparison to linear blocking I/O programming. In asynchronous programming, the codes become clumsier, and developers have to depend on the nested calls.

**Which is better?**

Node.js is a perfect framework when you want to develop a **server-side web application** like an online streaming platform. React.js is best suited when you want to **create a project with evolving states like dynamic inputs, buttons, etc.**

You can also use both the frameworks for a single project. You can develop back-end with Node.js framework and can use React.js for developing the front-end. The best example of such integration of frameworks is Netflix. Both frameworks have large and active community support. The choice between the two depends on your needs and requirements.

**What to learn next**

I hope this article cleared up the differences between Node.js and React.js. There is still a lot to learn to fully understand these tools and their uses.

To brush up on your React skills, check out Educative’s course [**Reintroducing React: V16 and Beyond**](https://www.educative.io/courses/reintroducing-react-v16-beyond). If you learned React a while back, it’s possible that you haven’t been up to date with every new feature/release. This course walks you through those updates with hands-on practice.