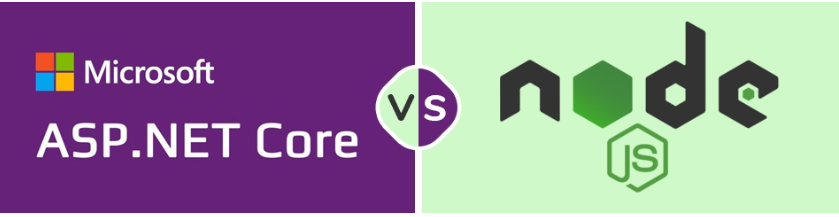
**.NET Core VS NodeJS**



In this topic, there’s no final victory here, as it depends upon the structure and the purpose of the application.

|  |  |  |
| --- | --- | --- |
| **Parameters** | **.NET Core** | **NodeJS** |
| **Developed By** | Microsoft | Ryan Dahl |
| **Release Year** | 2002 | 2009 |
| **Definition** | .NET core is an open-source part of the .NET Framework. It boasts **cross-platform functionality, micro services support**, docker containers possibilities, **high performance, and scalability** | Node.js is a single-threaded, **platform-independent**, open-source development framework that is mainly used to build **fast and scalable** server-side and network applications. It makes use of the V8 JavaScript runtime engine and utilizes an event-driven, non-blocking I/O architecture |
| **Scalability** | Both NodeJS and .NET Core are capable of seamlessly running on micro services architecture. **Also, both these frameworks are a perfect fit for the application to be run on distributed systems** | Node.js is perfectly designed for distributed systems. Micro services-based software enables components to scale autonomously which prevents an app from falling apart due to its weight. |
| **Platform compatibility** | It boasts cross-platform functionality, micro services support, docker containers possibilities, | Cross-platform: Linux, macOS, Windows |
| **Size of applications** | NET is a better choice for large apps | smaller and medium-sized apps, Node. js is the better option |
| **Security** | .NET Core is considered **more secure** due to its integrated Authorization, Authentication, HTTPS enforcement, and much more. | Node JS development has some incorporated security mechanisms, such as automatic vulnerability checking, logging, and monitoring, but requires third-party APIs to successfully and reliably configure them. As a result, loopholes can be created in the app, and the risk of a data breach can be increased |
| **Performance** | Compared to NodeJS, .NET Core is faster due to separate threads available for each user request, rather than creating the virtual ones in the background. | Due to the asynchronous model, Node.js apps have high performance without being too heavy in comparison to other solutions |
| **Community** | .NET **has more community** support on Stack Overflow | great support from its community and is more supported with the use of **GitHub**. |
| **Stability reliability and security** | **ASP.NET Core/.NET Core is definitely a winner in this category**. The security and reliability the platform provides make it a great option to create robust software with C# language | Node.js is **more reliable for complex enterprise software** developed with TypeScript than on its own. |
| **Full stack** | server-side framework , not full stack | Node.js has been regarded as a **full-stack JavaScript for serving both the client and the server-side** applications. |
| **Development Speed** | .NET Core demands less coding now, developers can easily optimize the code structure by means of writing much lesser statements. **.NET Core very close to Node.js in terms of development speed** | Node.js interprets the JavaScript code via **Google’s V8 JavaScript engine.** It compiles JavaScript code into the machine code(directly). This makes it **easier and faster to implement the code.** |
| **Concurrency** | .NET Core has an easier time working with CPU-intensive tasks and rendering static pages since the in-built IIS server kernel caching makes this process very straightforward | we can say that very load-intensive applications and a lot of incoming requests at the same time are **better handled by Node.js.** |
| **Resource Consumption** | Someone may think that the .NET core may be using only the old blocking thread pool method to handle requests, That's what I thought at the beginning. But **.Net core provides excellent support for using asynchronous programming patterns**. Net core also has an I/O completion method to handle asynchronous requests where the allocated threads are not waiting until the response arrives. NET core can also handle heavy resource-consuming requests like encoding and decoding videos and audio. .NET core gives a good competition when it comes to handling concurrent tasks. But as I said **node.js loses out to .NET when requests become CPU intensive.** | Node.js runs on a single thread so it uses fewer resources than traditional process request methods. What happens inside node.js request handling is it takes a request and if it can be processed immediately, sends the response. If it’s a time-consuming API call or long-running job pass it to the event loop and continue to the next incoming request. After the event loop has processed the request, the response is sent back using a callback function. |
| **Companies using Node.js** | Dell, Stack Overflow, Intel, Intuit, Cisco, Morgan Stanley, Siemens, and many more are using .NET Core for enterprise application **development for years.** | Netflix, PayPal,Medium, eBay. Node.js is very popular among **start-ups** as it allows you to build products faster and cheaper, especially on early stages of the product |

**Reference links**

<https://www.sam-solutions.com/blog/net-vs-node-js/>

<https://medium.com/c-sharp-progarmming/node-js-vs-net-core-the-winner-af461e19c2f>

<https://hackernoon.com/nodejs-vs-net-core---which-is-the-ultimate-server-side-development-platform>

<https://positiwise.com/blog/net-core-vs-node-js#:~:text=NET%20Core%20makes%20it%20more,for%20desktop%20and%20mobile%20applications.>

<https://medium.com/c-sharp-progarmming/node-js-vs-net-core-the-winner-af461e19c2f>

<https://www.tatvasoft.com/outsourcing/2022/03/asp-net-core-vs-node-js.html>

<https://intelvision.pro/blog/node-js-vs-net-core-what-to-choose-in-2022/>

<https://graffersid.com/net-core-vs-node-js/>