**Familiarize With NodeJS -** [Asish Raz](https://medium.com/@asishraz?source=post_page-----cb1be4b990c--------------------------------) Sep 25, 2021

A-Z of NodeJS

**NodeJS was written by Ryan Dahl in the year 2009. Check out his repo and don’t get fascinated with his commits.**

Before going into details for what exactly is NodeJS, let’s understand the key component here, which is ***JavaScript(JS)***, and then we will see why Node was written or what was the purpose behind launching Node.

Diagram

Description automatically generated

JavaScript

JS runs on the browser with the help of [**JSE**](https://en.wikipedia.org/wiki/JavaScript_engine)**(JavaScript Engine)**.  
JSE is nothing but a compiler that translated the JS code into machine code which browsers understand easily.

Diagram

Description automatically generated

JSE

And the browser which is not having JSE, *will not be able to run the JS code*. Sooner, different browsers started creating their own JSE to make it dynamic.

Diagram

Description automatically generated

JSE in different browsers

**Then, with all this how come NodeJS came into the picture?**

Diagram

Description automatically generated

early methodology

JS was getting used wisely in the browser, desktop, phone, and web-server, and to run it on the machine, there needs a runtime environment. And boommmm, NodeJS came as a savior.

**What exactly is NodeJS?**

*NodeJS is an*asynchronous*,*event-driven *JavaScript runtime platform that is used to build scalable network applications.****Asynchronous****— It executes the code without having any dependency and no order, which ends up improving the system efficiency.****Event-driven****— Flow determined by the events such as user actions, message passing, etc.*

Diagram

Description automatically generated

NodeJS

Diagram

Description automatically generated

NodeJS architecture

**How NodeJS works:**

Diagram

Description automatically generated

The working mechanism of NodeJS

Let’s say a client request *“Cricket Live Score”*to the server.  
Now the server can do the three things:

*1. Pass that request to the new server  
2. Pass that request to the database  
3. Pass that request to the file system.*

In case of passing the request to another server, it might take some time to get the response(let’s say 2 seconds)

Diagram

Description automatically generated

*For 1 client, it will take****2 seconds****and in the case of 5 clients, the 5th client will get the response in the****10th second****. There should be multiple request handling systems.*

***And we know, JS doesn’t support multiple-thread and Node has only a single thread.***

**The question arises, how does NodeJS then handle multiple clients/requests?**

Diagram

Description automatically generated

NodeJS with a single thread

**Then how do Non-Blocking I/O works?**

Diagram

Description automatically generated

Client **C1**request to the NodeJS web server.  
The single thread **T**will send that requests to the workers who work for the NodeJS server.

Now this thread **T**will not be blocked for **C1**and accessible to other clients as well.

Same like **C1**, another client **C2**request to Thread **T**and the same thing will happen.

This thread **T**will send that request to another worker  
And will not be blocked for the client **C2**and will be accessible to other clients as well

A picture containing text, indoor

Description automatically generated

**This is the way Non-blocking I/O works by not blocking itself for 1 client/request**

**What is the purpose of NodeJS?**

Initially, running JavaScript code outside the browser was next to impossible. Thanks to NodeJS, that we can now execute JS code outside the browser, which means on the server-side. And along with this, the main purpose of Node was to create dynamic web pages on a server.

**How does NodeJS make JS runs outside the browser?**

We all know that in beginning, Chrome has the V8 engine which is a JS engine, which allows JS code to execute. Now, the NodeJS replicated this engine on its infrastructure, and then it got the ability to understand the JS code as it has a V8 engine in it.  
Not only this, Node has a ‘**[libuv](https://github.com/libuv/libuv" \t "_blank)**’ library which is a way of asynchronously executing the code.

Diagram

Description automatically generated

Libuv

**Applications build on NodeJS:**

Graphical user interface

Description automatically generated with medium confidence

apps built on NodeJS

**History of NodeJS:**

**2009** — Created by ***Ryan Dahl***  
**2010**— [**NPM**](https://docs.npmjs.com/about-npm)Released [NPM- Node Package Manager; knows all the tree of dependencies of all the packages]  
**2011**— Big firm started adopting NodeJS [***LinkedIn, Uber***]  
**2013**— [**Ghost**](https://ghost.org/)— The first blogging platform completely on NodeJS  
**2014**— **io.js**— The major fork of *NodeJS*with *ES6*support  
**2015**— [**node.js**](https://openjsf.org/) foundation was born  
**2016**— [**YARN**](https://classic.yarnpkg.com/en/)released  
**2017**— **V8** started using NodeJS for the testing purpose  
**2018,19,20,21**— Major Versions Released

Now, I hope that you know the basics of NodeJS and how it works. With this, don’t just settle here, try exploring the project aspects and build some web apps.

If you are looking for a specific platform to have collective resources or links or projects to kick start with, you can follow this telegram channel:

**[NodeJS Learnings](https://t.me/nodejspadho" \t "_blank)**

[Dropping all relevant resources and my learnings towards NodeJS](https://t.me/nodejspadho" \t "_blank)

[t.me](https://t.me/nodejspadho" \t "_blank)

Or, if you are looking out for the resources in the web-development domain, join this channel as well: [***Yes, self-promotion is the foundation.***

**[towards web development](https://t.me/dumpingthoughts" \t "_blank)**

[Trying my best to provide all relevant resources in web development...](https://t.me/dumpingthoughts" \t "_blank)

[t.me](https://t.me/dumpingthoughts" \t "_blank)