Namaste Node.js

– By Akshay Saini

Season 01 →

Episode-10 → Thread Pool in libuv

Episode-11 → Creating a Server

Episode-12 → Databases - SQL & NoSQL

Episode-13 → Creating a database & mongoDB

For, Namaste Node.js Handwritten Pdf visit →

gitHub → rajeshjha2000

Episode-02 → JS on Server

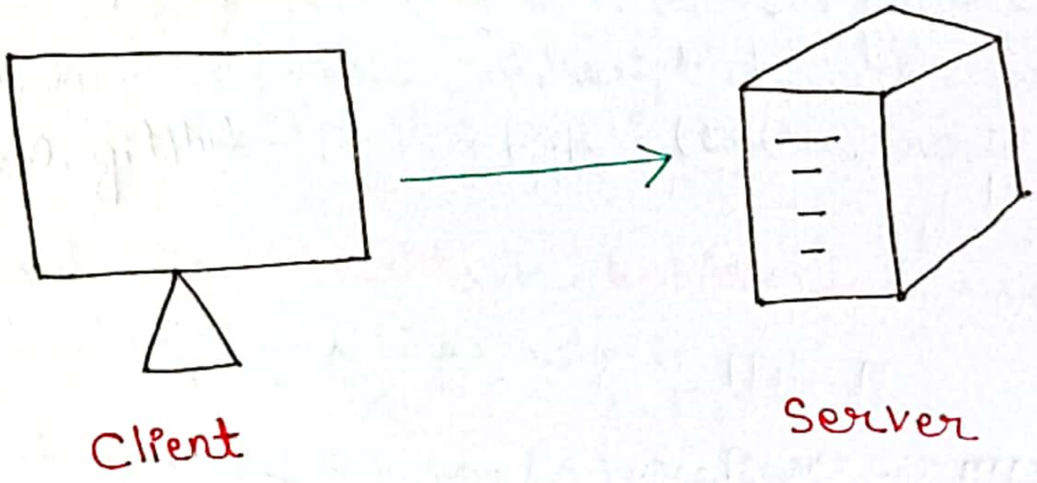
→ Node.js come with a philosophy that it can run outside the browser primarily on the server.

Q.> What is mean by server?

A> Server means nothing but a computer working remotely.

- Suppose if we are running a website on a server, it means we are running a website on a computer. So, it is one CPU or a machine which receiving that request. So that receiver or receiving computer is known as the server.

- When we try to google.com on client side, so every domain name maps to an IP address. This IP points to a device (server). So whenever we are requesting google.com that means this computer is making a request to the server.



- Initially, JS was the main language which was used inside the browser but after Node.js we can even run JS on Server.

- Now, this is very powerful because when JS was just used in frontend or on the web client. So, there were difference frontend developers and on the backend there were languages like Java, Python. So, there were separate backend developers.

- Now with JS coming on server as well as on web, it gave an opportunity for a single developer to develop full-stack.

- when JS can be run on both client & server it became a lot popular and a lot of developers now just have to learn one language.

- Node.js is C++ code.
- JS Engine - V8 (Google Chrome) is written in C++.
- Search - V8 GitHub, it's an open source.
  - C++ - 72.1 %, majority of the code of V8 is written in C++.

- At the end of the day, JS Engine is a C++ programme.
- Visit → "V8 JavaScript Engine" website, V8 is Google's open source high-performance JavaScript and Web Assembly engine, written in C++. It is used in Chrome and in Node.js.

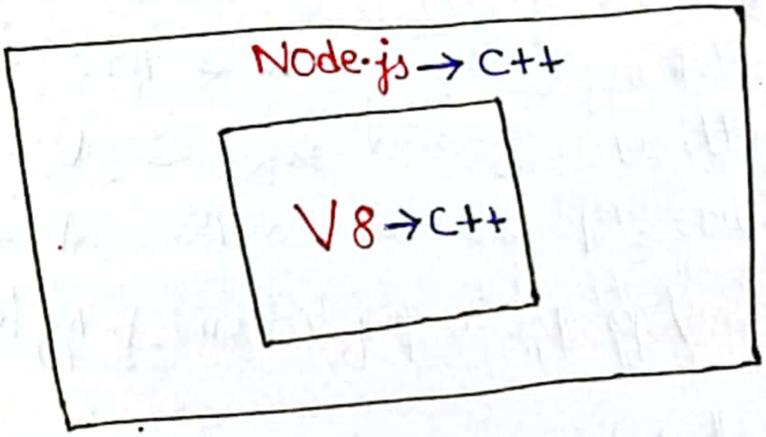
- It implements ECMAScripts and webAssembly, runs on windows, macOS and Linux system.

- V8 Can be embedded into any C++ application.
- The Job of V8 engine is to execute JS code, so what happens basically we write some JS code that JS code is read by V8 (C++ programme) and it is then converted a code which machine understands.

- JS is being read by C++ and then it is converted at the end in binary which computer understands.

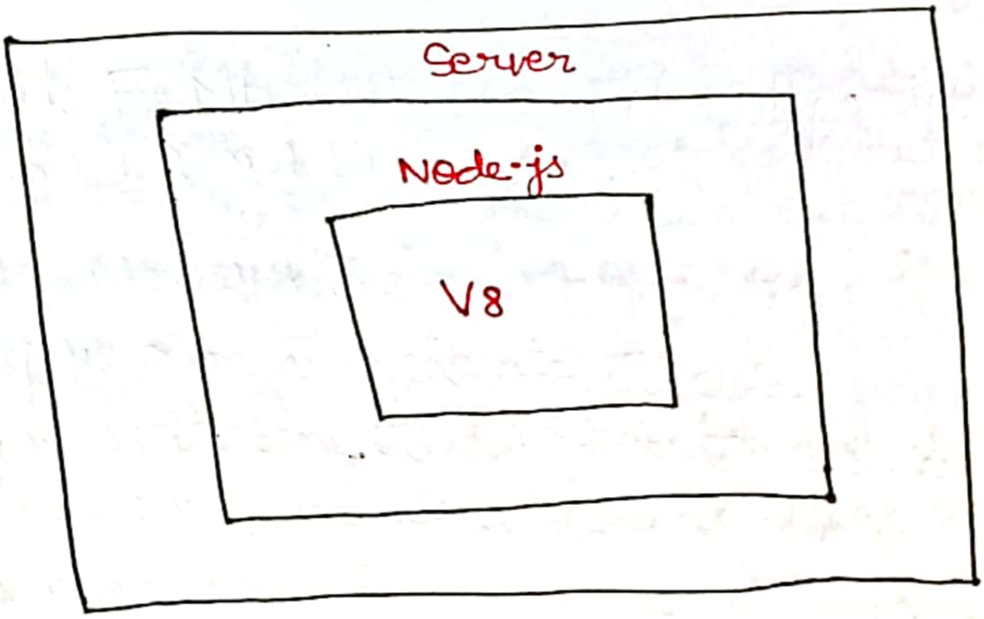
- "V8 Can be embedded into any C++ application" this is the main thing which led to the invention of Node.js.

- So what the Node.js Creator did, he created Node.js which is again a C++ application with V8 embedded into it.



Q:> But if V8 execute any piece of JS, why was there need of Node.js?

A:> When V8 can execute our JS code then just put V8 on the server and than V8 can execute every thing.



- V8 is a JS Engine that follows ECMAScript standards.

- ECMAScript is a standard for scripting languages including JavaScript, JS Script and Action Script.

- So, basically whatever JS, we write it follows a Standard.

  ex → Var a = 10;

The way we write JS that are maintained by a Standard that is known as ECMAScript.

- This standard is not only followed by JS only but a lot more languages follow this standard.

- So to run JS, the engine should understand the rules. ECMAScript defines the rules.

- JS engines have to follow these standards (rules).

- These standards are defined by a central committee who is responsible for adding new features into the language.

- All the JS Engines specifically follow the standards, they have to adhere to these ECMAScript standards, so that what we write in JS code, so any engine execute that JS code should give us the same result, it can only happen if all have same standards.

- All the JS Engines are built differently but at the end of the day what these JS Engine should do, it should take the JS code, converted into the code that machine understand.

- Every browser has their own implementation of the JS Engines but these JS Engine follows a specific format, this is known as ECMAScript standards.

- V8 has to follow ECMA script standards and it is written in C++ code, V8 can't go outside the ECMAScript Standards. So, Node.js gives extra superpower to V8 on top of it.

- Node.js is a C++ programme, it has V8 engine embedded in it and along with that there are more superpowers which it gives and that Node.js which it runs on the server makes it very powerful.

- Super Powers → APIs on Server

- If we want access Database onto our server, so V8 can't connect to Database, it doesn't have access to Database.

- Suppose if we want to make HTTP calls, V8 can't do that, only Job of V8 is to execute ECMAScript Standards.

- V8 can't connect to Database, it can't go to file system, it can't fetch image from the folder. So we need some super powers and those superpowers come in form of API and ~~to~~ that is core that Node.js adds onto V8 and it becomes powerful.

- V8 + SuperPowers → Node.js

And this is known as JS runtime.

- Node.js gitHub repository →
  - JS → 62.8%
  - C++ → 21.9%

- So, it is a C++ core programme but it has also a lot of JS APIs (superpowers), these API which user can call.

- If we are writing some code to connect Database that is written in JS, it contains JS + V8.

Q> "V8 is a C++ code", so what does it do and why it is a C++ code.

A> computer understands Binary Code, on top of this Binary code there is something known as Assembly code/language, on top of this assembly code there is Machine code, on top of machine code there is High Level language/code (C++).

- We user love to write JS because it's very easy. so this C++ code takes our JS code and converts it to Machine code (Low Level Code).

- So, we need JS Engine (C++ code) that converts our High level code (JS) to machine understandable code. This is the job of JS Engine.

- On the gitHub go to the Node.js repository, over there go to "deps" (dependencies) file.

so, there we will find V8 engine.

- JS Engine (V8) is just one dependency inside Node.js.
- ECMA Script Website - ECMA Specification, it is managed by "TC 39" community.
- It is a committee of 39 people who maintains it. It is like documentation for JS.

Follow, github → rajeshjha2000