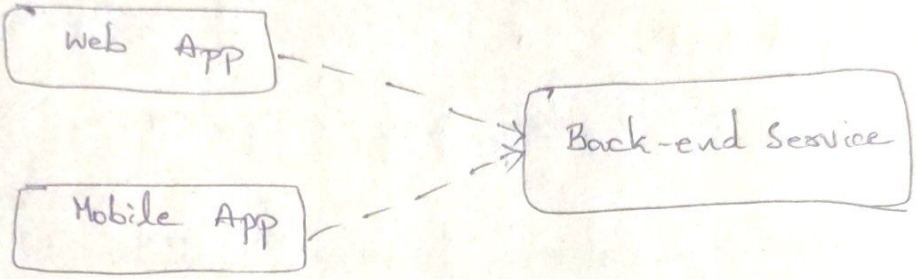
Node Js

(Mosh Hamedani) 23/07/2023

programmingwithmosh.com

Node Js: it a open source. A runtime environment for executing Javascript code out of the browsers.

→ We often use Node to build back-end services. also called APIs (Application Programming Interface)



→ Node is ideal for building highly-scalable, data-intensive and real-time apps.

Features:

* Great for prototyping and agile development.
* superfast and highly scalable.

Node App

Built twice as fast with fewer people

33% fewer lines of code

40% fewer files

2x request/sec

35% faster response time

* it uses javascript and is everywhere

* cleaner and more consistent codebase
* Large eco-system of open-source libs.

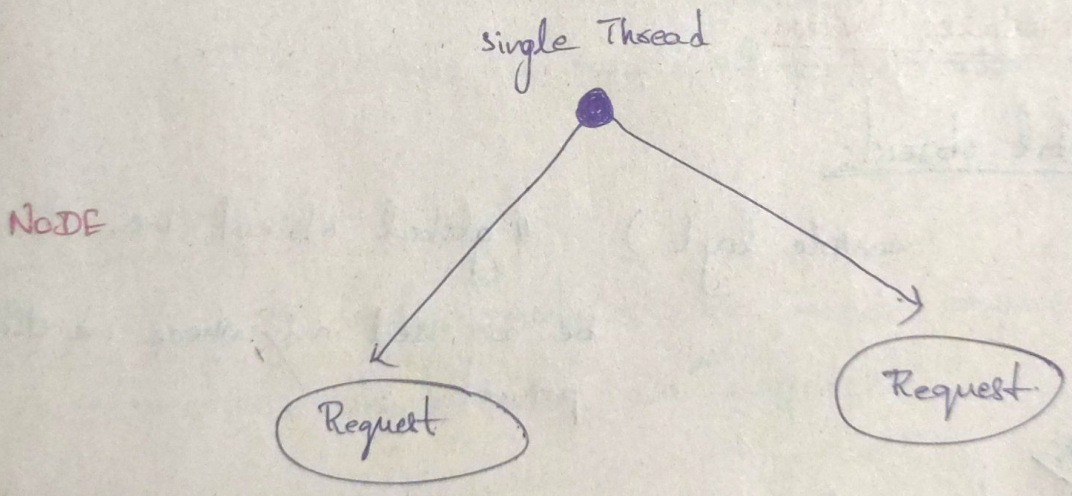
Node Architecture:

Ryan Dahl created the Node

Node is not a programming language or a framework it is a run-time environment

How Node Works

Node work with Non-blocking asynchoonous.



* Node applications are asynchronous by default.
* Do not use Node for CPU-intensive apps

* checking node version.

node --version ↵

output : V18.14.2

* mkdir NodeJs ↵
* cd NodeJs ↵
* NodeJs $ code .

→ to execute the code written in the file (app.js)

$ node app.js // is the file where we write the javascript code

output: Hello, Hemavath

Node Module System:

Global object:

console.log( ) // global object i.e, it can be accessed anywhere in the project.

Javascript:

setTimeout ( );
clearTimeout( );
setInterval( );
clearInterval( );

All these functions are belonging to the window object.

window.console.log( ).

In the node the window was replaced by the global

global.console.log( ).

in the node the variables, methods that are defined the global space are not attached to the global object.

```
var message = " ";
console.log(global.message);
```

output: undefined.

Modules:

* In node,
every file is a module

↳ console.log(module);

the variables and functions/methods declared in these modules are scoped to this particular module.

Loading a Module:

to load a module we need require() function, which takes an one argument.

Path Module:

require('path')

↳ a built-in module.

↳ const path = require('path');

```
var pathObj = path.parse(__filename)
console.log(pathObj);
```

Modules:

refer the example nodeJs folder Practice.

→ path
→ os
→ file system.
→ event (A signal that something has happened).
→ HTTP
→

To save the npm as dev dependencies.

$ npm i --save-dev nodemon ↵

Node JS Command line:

To read env:

NAME: HEMANTH PROFESSION = DEVELOPER node env.js

in env.js

```
console.log(process.env.NAME);
```

another way:

use npm

env.js

```
require('dotenv').config
```

$ npm i dotenv // and read the env's from the .env file

another way:

$ node -r dotenv/config env.js

REPL: Read Evaluate Print Loop

It is a node terminal

$ node ↵

```
> console.log('Hi');
Hi
undefined
>
```

```
> 5 === 5
true
>
```

argument.js

```
console.log(process.argv)

console.log(process.argv.slice(2)[0]);

process.argv.forEach((val, index) => {
    console.log(`${index}: ${val}`);
});
```

in terminal:

node argument.js name=HEMANTH

Date
24/07/2023

Introduction to NPM:

→ NPM is the standard package manager for Node.js

→ It has million packages listed in NPM registery

→ NPM is a way of download and manage dependencies for both fontend and backend applications using javascript.

Packages:

nodemon
axios
react-router-dom

→ npm run <task-name>
→ npm list
→ npm view <package-name> version
→ npm uninstall <package-name>
→ npm help

NPM (CLI commands)

→ npm init
→ npm install
→ npm install <package-name>
Eg, --save-dev, --no-save, --save-optional, --no-optional
→ npm install <package-name>@<version>
→ npm update
→ npm update <package-name>

→ To know what packages are installed globally.

```
$ npm root -g ↵
```

→ To run the executables which are in the node_modules bin folders

```
$ npm install cowsay ↵
$ npx cowsay I am Learning ↵
```

Semantic Versioning:

```
npm install express. ↵
```

package.json

```
{
  "dependencies": {
      "express": ^4.18.1
                 ^X.Y.Z
  }
}
```

X : The first digit is major version.

Y : The second digit is minor version.

Z : The third digit is patch version

^ : indicates that it changes the minor, patch versions on update, but not changing the major version.

~ : means change only patch version

JSON.stringify(tesla, null, 3);
JSON.stringify(tesla, undefined, 2);

* These 2 beautify the object as JSON object

path [Module]

filepath = "Users/Hemanth/codes";

path.dirname(filePath))

.basename( );

.extname( );

~~Path~~. join (path.dirname (filepath),
sampleFile);

Date
25/07/23

Rest API Convention:

(Small Project with Knowledge that learned).

CRUD Actions	HTTP Method	Endpoint
Get all movies.	GET	/api/movies
Get movie	GET	/api/movies/:id
Create movie	POST	/api/movies.
Update movie	PUT	/api/movies/:id
Delete movie	DELETE	/api/movies/:id

Date
20/03/23

Node.js

Node.js is an open-source, cross-platform javascript runtime environment.

provides all the necessary components in order to use and run a javascript program outside the browsers.

↓

until the node.js was introduced the javascript used run inside the browsers only.

↓

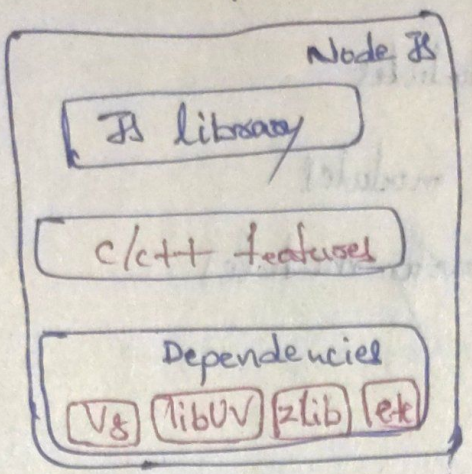
with node.js

javascript code can be run outside the browsers also.

With node.js:

→ traditional websites
→ Backend services like APIs
→ Real-time applications
→ Streaming services
→ CLI tools
→ Multiplayer games.

⇒ Node.js is a javascript runtime



Executing ways:

⇒ (1) Node REPL

R - Read
E - Evaluate
P - Print
L - Loop.

in the terminal:

node ↵ // it will open an interactive shell to read code

Ex:

```
console.log("Hello World"). ↵
```

Hello World.

```
> 2+2 ↵   // 4
> ctrl+c // quiet.
```

⇒ (2) Executing code in a javascript file in the command line

⇒

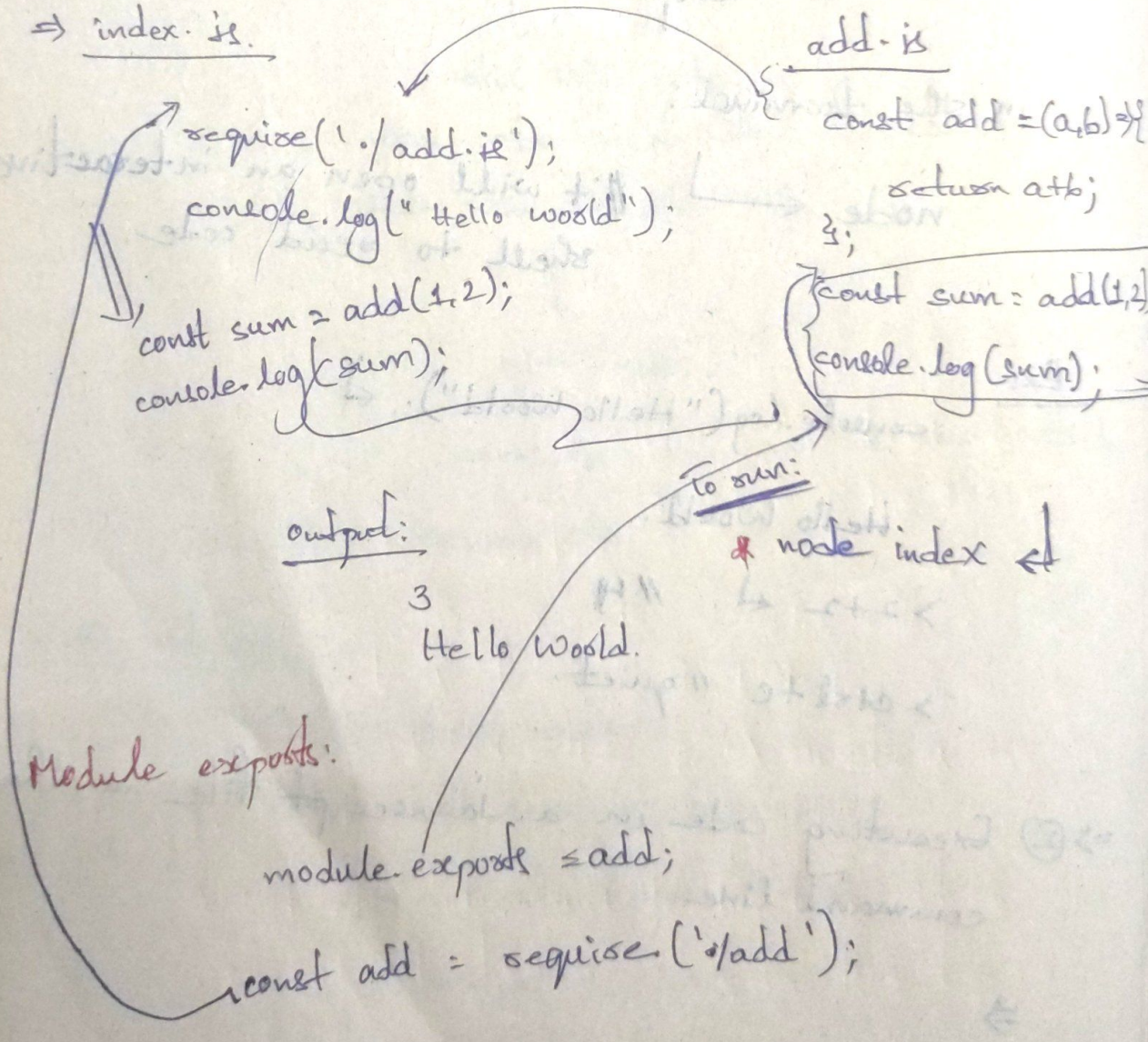
Types of Modules:

1) Local Modules
2) Built-in modules
3) Third-party modules

Local Modules:

CommonJs is a standard that states how a modu should be structured and shared.

Node.js adopted CommonJs when it started out and is what you will see in code bases.



iife.js

iife.js gives the private scope for each function

```
(function( ){
        const superHero = "Batman";
        console.log(superHero);
})( );

(function( ){
        const superHero = "Superman";
        console.log(superHero);
})( );
```

node iife ↵

25/07/202