12/06/2024

LEARNING NODE.JS

↳ prog with mosh YT vid

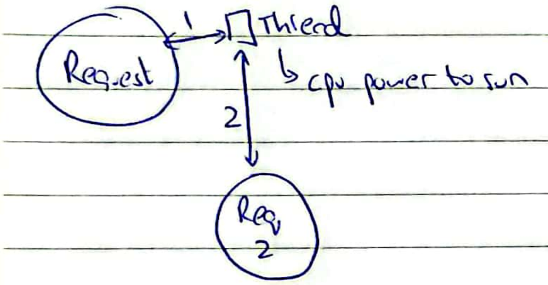
runtime env

for running JS → on server → uses same V8 JS engine as browser

usually only run on BROWSER

Node.js

→ Non Blocking & Asynchronous Architecture → not one by one, multiple requests can be handled together



→ one thread can handle multiple req's → makes Node.js highly scalable!

→ Node.js → used for data intensive, real time Apps

↳ not cpu intensive apps,

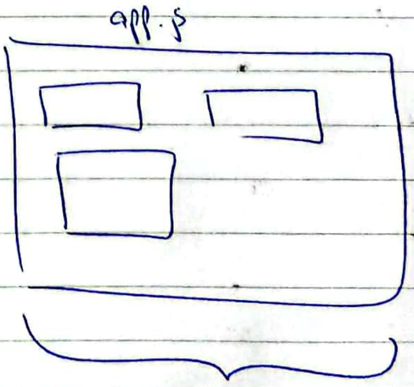
makes sense cause one cpu thread can get overloaded and the other requests are left waiting!

→ Node.js runs JS code

Node Module Sys

① OS
② File system
②Events
③ HTTP

* Avoid defining func / variable names in global scope



MODULE

every file is a module → variables/ func in that are of the scope!

```
console.log(module)
  OUTPUT: {
      id: ".",          →   d
      exports : {},     →   used to access diff (outside modules)
      filename : ' ',   →   complete path to file
      loaded : false,   →   boolean
      children : [],                  * will make sense later.
      paths : [

      ]
  }
```

* Node wraps the code inside the module, in the form of a function

(func(exports, require, module, _filename, _dirname){ } → NOT IMP!

NODE BUILT IN MODULES

① PATH -> Makes it easier to work with paths, rather than dealing with the path as a string

↳ root
dir
base → name of file where its called
ext : ".js" } → split into name & extension (from filename)
name : "app"

② OS → get total memory, free memory

③ FILE SYSTEM → always use asynchronous methods.

④ EVENT → EventEmitter } → CLASS → ∴ have to create object

↓ const EventEmitter = require('events');

↓ const emitter = new EventEmitter();

⑤ HTTP Module