1.What is Node Js Work. Why we use

Node JS is free open source JavaScript runtime environment and library through which we can run web applications outsite of brower

Node JS built on javascript so we use js as front end as well as backend so its open up development possibilities. It is easy to learn, cost effective, higher performance and improves response time

Node JS used in network applications, responsive web applications, client server applications]

Feature of Node Js

Single threaded, Asychronous, Open Source, Performance, Highly Scalable, Event Driven

2.How Node JS Work

Client send request to server through web application. Node js receive that request.

Request can be blocking type or not-blocking type. Node js add all request into loop queue.

And then pass one by one request to event loop. Then event loop check whether the request is simple and not require external resources if not require it create response and send back to client

If the request Is blocking a single thread from the thread pool is assign to complex request, and then that thread is responsible for the completing that particular request by accessing external resources. Onces task is completed response send back to the client

3. Why Node JS is single threaded

Node JS used single thread model in order to support async processing. With async processing an application can perform better and more scalable under web loads than multi thread applications

4. Event Driven Programming

Event driven programming heavily make uses of events for triggering various functions. An event can be anything like mouse click, key press etc. when event occur callback function attach to event will be executed

5. What is I/O operations

I/O operations refer to reading/writing file or network operation to get data from external into application or send data from your application to external

6. What is event loop

Event loop is endless loop, which wait for task, executes them, and then sleeps until next task is receive

Event loop excutes tasks from event queue only when call stack is empty

Event loop excutes tasks starting from the oldest first

I will explain it through one example

So I have one block of code at starting I have one console.log statement following the setTimeOut() functionality and at last console.log statement so when code excuection started

The first console.log() pushed into the call stack and statement of console logged into console. And then task is pop out from the stack.

The setTimeOut encounter by javascript engine and pushed into the callstack now setTimeOut is register in web api with timer feature attach to it and start timer of 5000 ms also callback function attach to it

Now javscript move to next line without waiting for anything and setTImeOut remove from the callstack

Another console.log is encounter by javascript engine and it push into call stack after execution finished statement log into console

So now after 5000 ms later, after the timer expiration the callback function move into event queue through call stack for execution. So callback is excutes in console statement logged in and after it is popped out

7. Types of APIs Node Js

Synchronous API : use for blocking functionality

Asynchronous API : use of non-blocking functionality

8. What is Control Flow Statement/Functions

Control flow statement/ functions are piece of code that runs between the several asynchronous calls

9. What is Callback

Callback functions are functions executes after a given task. Its allows other code to run in the meantime and prevent other code from blocking. Being a asynchronous platform, Node JS heavily relies on callback. All API of node written to support callbacks.

--

Callback function make sure that function will not excutes until task in completed but will run after task is completed. It helps us to asynchronous code and keep up safe away from error

10. What is Callback Hell

Callback hell is situation in which we have complex nested callbacks. In which each callback takes arguments that have been obtained as result of previous callback. So this leads lesser readability of code and maintainability

We can avoid callback hell with help of promises and async await

11. Error First Call Back

In Node JS function either return an error object or any successful data return by function

The first argument in the function is reserverd for error object

The second argument in the function is reserved for successful data return by function

12. Node Js Child thread

Node Js is single thread language . so to created multiple thread using child thread/ process child\_process module

It can be created in 4 way spawn, fork, exec execute and then we have execFile execute file

13. What are Modules

Modules are like javascript library that can be used in node js applications to include set of functions with help of require() function with the parentheses containing module name

14. EventEmitter

EventEmitter is used to handle custom event in Node JS

EventEmitter holds all objects that can emit events

So whenever object from EventEmitter class thrown an event, all attached function are called upon synchronously

15. What is REPL

REPL is stand for Read, Eval, Print and Loop. It is similar to windows console, in which we enter command, then system respond with an output

16. What is Middleware

Middleware functions are functions that have access request and response object and moving to next middleware function in the application’s request-response cycle.

Middleware get excuted after server recheive request and before the controller send the response

Middleware can process request multiple times before the server work on request

App.get( path , (req, res, next) => {} , (res, res) => {} )

Middleware takes three arguments req object , res object and next function in application of request-response cycle

They are also capable of ending the cycle by sending a response when condition is satisfied if condition is not satisfied they start executing next function in the stack

Middleware functions can perform following task

Excutes any code

Make changes in request and response objects

End the request response cycle

Call the next middleware in the stack

17. What is Express Js

Express Js is routing and middleware framework for handling different routing of the web pages and work between the request and response life cycle

18. Diff HTTP requests

HTTP protocols is used to enable communication between client and servers.

HTTP protocol work as request response protocol between the client and server

Get Method :-

Get is used to request data from server.

Get data is send in header and only limited amount of data can be data.

Get data can be bookmarked.

Post Method :-

Post is send data to the server to create/update resource.

Post data send in body and there is no limitation in sending data.

Post data can be bookmarked.

Put Method :-

Put Method is used to update the resource on server

Patch Method : -

Patch is used to partial modification of the resource

Delete Method :-

Delete is used to delete resource from the server

18. How to secure Node Js applications

Helmet is middleware is used to make secure node js applications