Most asked Node.JS Interview Questions & Answers (Freshers)

Node.js is one of the most popular runtime environments in the world, known for its efficiency, scalability, and ability to handle asynchronous operations. It is built on Chrome's V8 JavaScript engine for executing JavaScript code outside of a browser. It provides an event-driven, non-blocking (asynchronous) I/O, and cross-platform runtime environment for building highly scalable server-side applications using JavaScript. Node.js is extensively used by top companies such as **LinkedIn, Netflix, Walmart, Uber, PayPal, NASA**, and many more because of its robust features and performance.

```
import "../index.css";
import { ReactComponent as ArrowIcon } from "../assets/icons/arrow.svg";
import { ReactComponent as BoltIcon } from "../assets/icons/bolt.svg";
import { ReactComponent as RightArrowIcon } from "../assets/icons/right-arrow.svg";

import React, { useState, useEffect, useRef } from "react";
import { CSSTransition } from "react-transition-group";
```

```
</ /> "eslintConfig": {
  "extends": [
    "react-app",
    "react-app/
```

```
import React, { useEffect } from "react";
import { CSSTransition } from "react-transition-group";
import "../../index.css";
import "./modal.css";

const Modal = (props) => {
 //set close by state

 //set close by "esc"
 const closeOnEscapeKeyDown = (e) => {
  if (e.charCode || e.keyCode === 27) {
   props.onClose();
  }
 };

 //use effect to detect click "esc"
 useEffect(() => {
  document.body.addEventListener("keydown", closeOnEscapeKeyDown);
  return function cleanup() {
   document.body.removeEventListener("keydown", closeOnEscapeKeyDown);
  };
 }, []);
```

1. What is Node.js?

Node.js is a JavaScript engine used for executing JavaScript code outside the browser. It is normally used to build the backend of the application and is highly scalable.

2. What is the difference between Node.js and JavaScript?

JavaScript is a scripting language whereas Node.js is an engine that provides the runtime environment to run JavaScript code.

3. Is Node.js single-threaded?

Yes, Node.js is single-threaded by default. However, it utilizes event-driven architecture and non-blocking I/O operations to handle multiple concurrent requests efficiently, enabling scalability and high performance in applications.

4. What kind of API function is supported by Node.js?

There are two types of API functions supported by Node.js:

- **Synchronous:** These API functions are used for blocking code.
- **Asynchronous:** These API functions are used for non-blocking code.

5. What is a module in Node.js?

In Node.js Application, a Module can be considered as a block of code that provides a simple or complex functionality that can communicate with external applications. Modules can be organized in a single file or a collection of multiple files/folders. Modules are useful because of their reusability and ability to reduce the complexity of code into smaller pieces. Some examples of modules are. http, fs, os, path, etc.

6. What is npm and its advantages?

npm (Node Package Manager) is the default package manager for Node.js. It allows developers to discover, share, and reuse code packages easily. Its advantages include dependency management, version control, centralized repository, and seamless integration with Node.js projects.

7. What is middleware?

Middleware is the function that works between the request and the response cycle. Middleware gets executed after the server receives the request and before the controller sends the response.

8. How does Node.js handle concurrency even after being single-threaded?

Node.js handles concurrency by using asynchronous, non-blocking operations. Instead of waiting for one task to complete before starting the next, it can initiate multiple tasks and continue processing while waiting for them to finish, all within a single thread.

9. What is control flow in Node.js?

Control flow in Node.js refers to the sequence in which statements and functions are executed. It manages the order of execution, handling asynchronous operations, callbacks, and error handling to ensure smooth program flow.

10. What do you mean by event loop in Node.js?

The event loop in Node.js is a mechanism that allows it to handle multiple asynchronous tasks concurrently within a single thread. It continuously listens for events and executes associated callback functions.

11. What is the order in which control flow statements get executed?

- The order in which the statements are executed is as follows:
- Execution and queue handling
- Collection of data and storing it
- Handling concurrency
- Executing the next lines of code

12. What are the main disadvantages of Node.js?

Here are some main disadvantages of Node.js listed below:

Single-threaded nature: May not fully utilize multi-core CPUs, limiting performance.

NoSQL preference: Relational databases like MySQL aren't commonly used.

Rapid API changes: Frequent updates can introduce instability and compatibility issues.

13. What is REPL in Node.js?

REPL in Node.js stands for Read, Evaluate, Print, and Loop. It is a computer environment similar to the shell which is useful for writing and debugging code as it executes the code on go.

14. How to import a module in Node.js?

We use the require module to import the External libraries in Node.js. The result returned by require() is stored in a variable which is used to invoke the functions using the dot notation.

15. What is the difference between Node.js and AJAX?

Node.js is a JavaScript runtime environment that runs on the server side whereas AJAX is a client-side programming language that runs on the browser.

16. What is package.json in Node.js?

package.json in Node.js is a metadata file that contains project-specific information such as dependencies, scripts, version, author details, and other configuration settings required for managing and building the project.

17. What is the most popular Node.js framework used these days?

The most famous Node.js framework used is Express.js as it is highly scalable, efficient, and requires very few lines of code to create an application.

18. What are promises in Node.js?

A promise is basically an advancement of callbacks in NodeJS. In other words, a promise is a JavaScript object which is used to handle all the asynchronous data operations. While developing an application you may encounter that you are using a lot of nested callback functions which causes a problem of callback hell. Promises solve this problem of callback hell.

19. What is event-driven programming in Node.js?

Event-driven programming is used to synchronize the occurrence of multiple events and to make the program as simple as possible. The basic components of an Event-Driven Program are:

- A callback function (called an event handler) is called when an event is triggered.
- An event loop that listens for event triggers and calls the corresponding event handler for that event.

20. What is a buffer in Node.js?

The Buffer class in Node.js is used to perform operations on raw binary data. Generally, Buffer refers to the particular memory location in memory. Buffer and array have some similarities, but the difference is array can be any type, and it can be resizable. Buffers only deal with binary data, and it can not be resizable. Each integer in a buffer represents a byte. console.log() function is used to print the Buffer instance.

21. What are streams in Node.js?

Streams are a type of data-handling method and are used to read or write input into output sequentially. Streams are used to handle reading/writing files or exchanging information in an efficient way. The stream module provides an API for implementing the stream interface. Examples of the stream object in Node.js can be a request to an HTTP server and process.stdout are both stream instances.

22. Explain crypto module in Node.js

The crypto module is used for encrypting, decrypting, or hashing any type of data. This encryption and decryption basically help to secure and add a layer of authentication to the data. The main use case of the crypto module is to convert the plain readable text to an encrypted format and decrypt it when required.

23. What is callback hell?

Callback hell is an issue caused due to a nested callback. This causes the code to look like a pyramid and makes it unable to read To overcome this situation we use promises.

24. Explain the use of timers module in Node.js

The Timers module in Node.js contains various functions that allow us to execute a block of code or a function after a set period of time. The Timers module is global, so we do not need to use require() to import it.

It has the following methods:

- setTimeout() method
- setImmediate() method
- setInterval() method

25. Difference between setImmediate() and process.nextTick() methods

The process.nextTick() method is used to add a new callback function at the start of the next event queue. it is called before the event is processed. The setImmediate is called at the check phase of the next event queue. It is created in the poll phase and is invoked during the check phase.

26. What is the difference between setTimeout() and setImmediate() method?

The setImmediate function is used to execute a particular script immediately whereas the setTimeout function is used to hold a function and execute it after a specified period of time.

27. What is the difference between the spawn() and fork() method?

Both these methods are used to create new child processes; the only difference between them is that the spawn() method creates a new function that Node runs from the command line whereas the fork() function creates an instance of the existing fork() method and creates multiple workers to perform on the same task.

28. Explain the use of passport module in Node.js

The passport module is used for adding authentication features to our website or web app. It implements authentication measures which helps to perform sign-in operations.

29. What is fork in Node.js?

Fork is a method in Node.js that is used to create child processes. It helps to handle the increasing workload. It creates a new instance of the engine which enables multiple processes to run the code.

30. What are the three methods to avoid callback hell?

The three methods to avoid callback hell are:

- Using async/await()
- Using promises
- Using generators

31. What is body-parser in Node.js?

Body-parser is the Node.js body-parsing middleware. It is responsible for parsing the incoming request bodies in a middleware before you handle it. It is an NPM module that processes data sent in HTTP requests.

32. What is CORS in Node.js?

The word CORS stands for “Cross-Origin Resource Sharing”. Cross-Origin Resource Sharing is an HTTP-header based mechanism implemented by the browser which allows a server or an API to indicate any origins (different in terms of protocol, hostname, or port) other than its origin from which the unknown origin gets permission to access and load resources. The cors package available in the npm registry is used to tackle CORS errors in a Node.js application.

33. Explain the tls module in Node.js?

The tls module provides an implementation of the Transport Layer Security (TLS) and Secure Socket Layer (SSL) protocols that are built on top of OpenSSL. It helps to establish a secure connection on the network.

34. What is a cluster in Node.js?

Due to a single thread in node.js, it handles memory more efficiently because there are no multiple threads due to which no thread management is needed. Now, to handle workload efficiently and to take advantage of computer multi-core systems, cluster modules are created that provide us the way to make child processes that run simultaneously with a single parent process.

35. Explain some of the cluster methods in Node.js

- **Fork():** It creates a new child process from the master. The isMaster returns true if the current process is master or else false.
- **isWorker:** It returns true if the current process is a worker or else false.
- **process:** It returns the child process which is global.
- **send():** It sends a message from worker to master or vice versa.
- **kill():** It is used to kill the current worker.

36. How to manage sessions in Node.js?

Session management can be done in node.js by using the express-session module. It helps in saving the data in the key-value form. In this module, the session data is not saved in the cookie itself, just the session ID.

37. Explain the types of streams in Node.js

Types of Stream:

- **Readable stream:** It is the stream from where you can receive and read the data in an ordered fashion. However, you are not allowed to send anything. For example, fs.createReadStream() lets us read the contents of a file.
- **Writable stream:** It is the stream where you can send data in an ordered fashion but you are not allowed to receive it back. For example, fs.createWriteStream() lets us write data to a file.
- **Duplex stream:** It is the stream that is both readable and writable. Thus you can send in and receive data together. For example, net.Socket is a TCP socket.
- **Transform stream:** It is the stream that is used to modify the data or transform it as it is read. The transform stream is basically a duplex in nature. For example, zlib.createGzip stream is used to compress the data using gzip.

38. How can we implement authentication and authorization in Node.js?

Authentication is the process of verifying a user's identity while authorization is determining what actions can be performed. We use packages like Passport and JWT to implement authentication and authorization.

39. Explain the packages used for file uploading in Node.js?

The package used for file uploading in Node.js is Multer. The file can be uploaded to the server using this module. There are other modules in the market but Multer is very popular when it comes to file uploading. Multer is a node.js middleware that is used for handling multipart/form-data, which is a mostly used library for uploading files.

40. Explain the difference between Node.js and server-side scripting languages like Python

Node.js is the best choice for asynchronous programming Python is not the best choice for asynchronous programming. Node.js is best suited for small projects to enable functionality that needs less amount of scripting. Python is the best choice if you're developing larger projects. Node.js is best suited for memory-intensive activities. Not recommended for memory-intensive activities. Node.js is a better option if your focus is exactly on web applications and website development. But, Python is an all-rounder and can perform multiple tasks like- web applications, integration with back-end applications, numerical computations, machine learning, and network programming. Node.js is an ideal and vibrant platform available right now to deal with real-time web applications. Python isn't an ideal platform to deal with real-time web applications. The fastest speed and great performance are largely due to Node.js being based on Chrome's V8 which is a very fast and powerful engine. Python is slower than Node.js, As Node.js is based on fast and powerful Chrome's V8 engine, Node.js utilizes JavaScript interpreter. Python using PyPy as Interpreter. In case of error handling and debugging Python beats Node.js. Error handling in Python takes significantly very little time and debugging in Python is also very easy compared to Node.js.