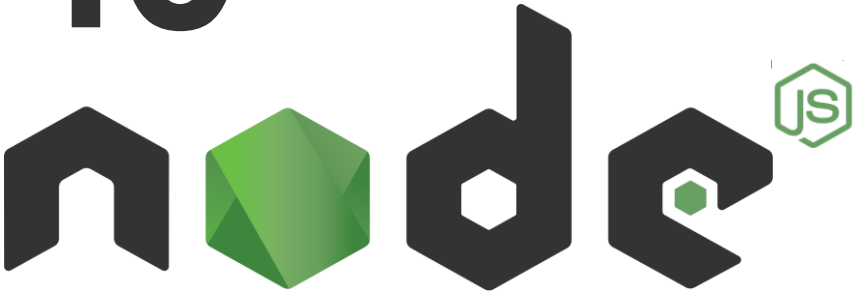
40



INTERVIEW QUESTIONS

The Key Concepts To Master

Follow for more insights:
MUHAMMAD SALMAN HUSSAIN

40 node INTERVIEW QUESTIONS

The Key Concepts To Master

1. What is Node.js?
Answer: Node.js is a JavaScript engine used for executing JavaScript code outside the browser, commonly used to build scalable backend applications.

2. What is the difference between Node.js and JavaScript?
Answer: JavaScript is a scripting language, while Node.js is a runtime environment that allows JavaScript to run on the server side.

3. Is Node.js single-threaded?
Answer: Yes, Node.js is single-threaded but uses event-driven architecture and non-blocking I/O to handle multiple requests efficiently.

4. What kind of API function is supported by Node.js?
Answer: Node.js supports both synchronous (blocking) and asynchronous (non-blocking) API functions.

5. What is a module in Node.js?
Answer: A module in Node.js is a block of code that provides specific functionality, which can be reused across different parts of an application.

Follow for more insights:
MUHAMMAD SALMAN HUSSAIN

6. What is npm and its advantages?
Answer: npm is the default package manager for Node.js, offering benefits like dependency management, version control, and a centralized repository.

7. What is middleware?
Answer: Middleware functions execute between the request and response cycle, performing tasks like logging, authentication, and data processing.

8. How does Node.js handle concurrency despite being single-threaded?
Answer: Node.js handles concurrency through asynchronous, non-blocking operations, allowing multiple tasks to run simultaneously within a single thread.

9. What is control flow in Node.js?
Answer: Control flow refers to the order in which code statements and functions are executed, managing asynchronous operations and error handling.

10. What do you mean by event loop in Node.js?
Answer: The event loop is a mechanism that processes asynchronous tasks in a single thread by continuously checking for and executing callback functions.

11. What are the main disadvantages of Node.js?
Answer: Disadvantages include its single-threaded nature, preference for NoSQL databases, and rapid API changes that can cause instability.

12. What is REPL in Node.js?
Answer: REPL stands for Read, Evaluate, Print, and Loop; it's an interactive environment for executing Node.js code and debugging.

13. How to import a module in Node.js?
Answer: Use the `require()` function to import external modules, storing the result in a variable for use in the application.

14. What is the difference between Node.js and AJAX?
Answer: Node.js is a server-side runtime environment, while AJAX is a client-side technique for asynchronously updating parts of a web page.

15. What is package.json in Node.js?
Answer: `package.json` is a metadata file in Node.js that contains information about the project, such as dependencies, scripts, and version.

16. What is the most popular Node.js framework used these days?
Answer: The most popular Node.js framework is Express.js, known for its scalability and minimalistic approach to building web applications.

17. What are promises in Node.js?
Answer: Promises in Node.js are objects that handle asynchronous operations, providing a cleaner alternative to callback functions.

18. What is event-driven programming in Node.js?
Answer: Event-driven programming synchronizes multiple events using event loops and callback functions to simplify program flow.

19. What is buffer in Node.js?
Answer: A buffer is a temporary storage space for binary data, allowing Node.js to handle raw data directly.

20. What are streams in Node.js?
Answer: Streams are objects used to handle continuous data flows, allowing for efficient reading and writing of data.

21. Explain crypto module in Node.js.
Answer: The crypto module provides cryptographic functionality, such as encryption, decryption, and hashing of data.

22. What is callback hell?
Answer: Callback hell refers to the problematic situation caused by deeply nested callbacks, making code difficult to read and maintain.

23. Explain the use of timers module in Node.js.
Answer: The timers module allows execution of code after a specified delay or immediately in the next event loop cycle using functions like `setTimeout()` and `setImmediate()`.

24. What is the difference between setImmediate() and process.nextTick() methods?
Answer: `process.nextTick()` executes callbacks at the start of the next event loop, while `setImmediate()` executes them at the end of the current event loop.

25. What is the difference between setTimeout() and setImmediate() method?
Answer: `setTimeout()` schedules a callback after a specified delay, whereas `setImmediate()` executes it immediately after I/O events.

26. What is the difference between spawn() and fork() method?
Answer: `spawn()` runs a new process from the command line, while `fork()` creates a new instance of the existing process to perform parallel tasks.

27. Explain the use of passport module in Node.js.
Answer: The passport module adds authentication features to applications, supporting various sign-in methods.

28. What is fork in Node.js?
Answer: Fork is a method to create child processes that allow parallel execution of tasks in Node.js.

29. What are the three methods to avoid callback hell?
Answer: To avoid callback hell, use `async/await`, promises, or generators.

30. What is body-parser in Node.js?
Answer: Body-parser is middleware that parses incoming request bodies in a middleware before handling it in Node.js applications.

31. What is CORS in Node.js?
Answer: CORS stands for Cross-Origin Resource Sharing, allowing restricted resources on a web page to be requested from another domain.

32. Explain the tls module in Node.js.
Answer: The tls module provides an implementation of TLS and SSL protocols to establish secure network connections.

33. What is a cluster in Node.js?
Answer: A cluster allows Node.js to utilize multiple cores of a machine by creating child processes that share the same server port.

34. How to manage sessions in Node.js?
Answer: Sessions in Node.js can be managed using the `express-session` module, which stores session data on the server.

35. Explain the types of streams in Node.js.
Answer: Types of streams include readable, writable, duplex (both), and transform (modifies data) streams.

36. How can we implement authentication and authorization in Node.js?
Answer: Use packages like Passport for authentication and JWT for managing tokens to implement security in Node.js applications.

37. Explain the packages used for file uploading in Node.js.
Answer: Multer is a popular middleware used for handling file uploads in Node.js.

38. How to handle database connection in Node.js?
Answer: Database connections in Node.js are managed using drivers like MySQL and libraries like Mongoose for MongoDB.

39. How to read command line arguments in Node.js?
Answer: Use the `process.argv` array to access command-line arguments passed when running a Node.js application.

40. What are child processes in Node.js?
Answer: Child processes allow Node.js to handle multiple tasks concurrently by creating subprocesses that can run independently.

Download or save this document for later.

Follow for more insights:
MUHAMMAD SALMAN HUSSAIN