ARYATECHNO

Node.js Interview Questions and Answers pdf

Topics : Node js
Written on December 06, 2022

Node.js Interview Questions and Answers

1. which the framework most commonly is used in node.js?

- Express

2. Define the steps by which you can async in Node.js?

By following steps you can async Node.js

1. First class functions
2. Function composition
3. Callback Counters
4. Event loops

3. Where can we use node.js?

- Real-time back end Web applications
- Network applications
- Distributed systems
- General purpose applications

4. What are the tasks that should be done asynchronously using the event loop?

- I/O operations
- Heavy computation
- Anything requiring blocking

5. Why code written in Node.JS is pretty fast although being written in JavaScript?

Node.js is built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.

6. What is an event loop in Node.js?

An event loop in Node.js is used to process and handle external events and to convert them into callback invocations an event loop is used. So, at I/O calls, node.js can switch from one request to another.

7. A stream fires data event when there is data available to read.

A stream fires data event when there is data available to read.

8. How Node.js overcomes the problem of blocking of I/O operations?

Node.js solves this problem by putting the event based model at its core, using an event loop instead of threads.

9. What is package.json?

package.json is present in the root directory of any Node application/module and is used to define the properties of a package. It can also be used to update dependencies of a Node application.

10. What does it mean “non-blocking” in node.js?

In node.js “non-blocking” means that its IO is non-blocking. Node uses “libuv” to handle its IO in a platform-agnostic way. On windows, it uses completion ports for unix it uses epoll or kqueue etc. So, it makes a non-blocking request and upon a request, it queues it within the event loop which call the JavaScript ‘callback’ on the main JavaScript thread.

11. Which method of fs module is used to remove a directory?

fs.rmdir(path, callback) is the method which is used to remove a directory.

12. What is ‘Callback’ in node.js?

Callback function is used in node.js to deal with multiple requests made to the server. Call back function allows the server to deal with pending request first and call a function when it is finished.

13. Which code can make a request to a web server?

http.request(options, callback) method can be used to make a request a web server.

14. What are the pros and cons of Node.js?

Pros:

- If your application does not have any CPU intensive computation, you can build it in Javascript top to bottom, even down to the database level if you use JSON storage object DB like MongoDB.
- Crawlers receive a full-rendered HTML response, which is far more SEO friendly rather than a single page application or a websockets app run on top of Node.js.

Cons:

- Any intensive CPU computation will block node.js responsiveness, so a threaded platform is a better approach.
- Using relational database with Node.js is considered less favourable.

15 . Which of the following code prints process version?

process.version can be used to get the current process version.

16. What are the two arguments that async.queue takes?

- Task function
- Concurrency value

17. What is Features of Node.js?

- Asynchronous and Event Driven − All APIs of Node.js library are asynchronous, that is, non-blocking.

- Very Fast − Being built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.

- Single Threaded but Highly Scalable − Node.js uses a single threaded model with event looping.

- No Buffering − Node.js applications never buffer any data. These applications simply output the data in chunks.

- License − Node.js is released under the MIT license

18. Why Node.js is single threaded?

For async processing, Node.js was created explicitly as an experiment. It is believed that more performance and scalability can be achieved by doing async processing on a single thread under typical web loads than the typical thread based implementation.

19. What are the two types of API functions in Node.js?

- Asynchronous, non-blocking functions
- Synchronous, blocking functions

20. Is it free to use Node.js?

Yes! Node.js is released under the MIT license and is free to use.

21. What do you mean by the term I/O?

I/O is the shorthand for input and output, and it will access anything outside of your application. It will be loaded into the machine memory to run the program, once the application is started.

22. What is npm?

npm stands for Node Package Manager. npm is used to install node js module/package

23. What is global installation of dependencies?

Globally installed packages/dependencies are stored in **<user-directory>**/npm directory. Such dependencies can be used in CLI (Command Line Interface) function of any node.js but can not be imported using require() in Node application directly. To install a Node project globally use -g flag.

```
D:\Nodejs>npm install express -g
```

24. What is local installation of dependencies?

By default, npm installs any dependency in the local mode. Here local mode refers to the package installation in node_modules directory lying in the folder where Node application is present. Locally deployed packages are accessible via require(). To install a Node project locally following is the syntax.

```
D:\Nodejs>npm install express
```

25. What does event-driven programming mean?

In computer programming, event driven programming is a programming paradigm in which the flow of the program is determined by events like messages from other programs or threads. It is an application architecture technique divided into two sections 1) Event Selection 2) Event Handling.

26. What is Event Emmitter?

EventEmitter class lies in **events** module.

```
//import events module
var events = require('events');

//create an eventEmitter object
var eventEmitter = new events.EventEmitter();
```

EventEmitter provides multiple properties like **on** and **emit**. **on** property is used to bind a function with the event and **emit** is used to fire an event.