

Top 10 Node Js Interview Question



Node Js Interview Questions

Node.js is a framework that acts like a server-side platform that is built on Google's Javascript engine. It is open-source software and hence can be used for free. It uses a non-blocking I/O model that is lightweight and can run across distributed services. It helps in the development of server-side and networking applications. It has many libraries consisting of JavaScript modules which make development easier.

Every interview is different, and the scope of a job is different too. Keeping this in mind, we have shared the most common Node.js Interview Questions and Answers to help you get success in your interview.

Let us have a look at the 2022 Node.js Interview Questions that are being asked in interviews.

Q1) What is Node.js, and explain its features?

Node.js is a runtime platform built on Google Chrome's JavaScript engine. It is a single thread model that uses the concurrency model for its events to be looped. Instead of blocking an application, it helps in registering a callback to the new application and allows the present application to continue. That results in the handling of concurrent operations without creating multiple threads of execution. It uses JavaScript with C or C++ for interacting with a filesystem. The main features of node.js are:

- **Node.js library:** All developers are mostly already comfortable with JavaScript. Node.js has a library built over JavaScript. Hence developers find it easy to use node.js.
- **Single-Threaded and highly scalable:** It uses a single thread for event looping. Though the responses may not reach the server on time, this does not block any operations. The normal servers have limited threads to handle the requests, and Node.js creates a single thread to handle a large number of requests.
- **No Buffer:** These applications do not need any buffer and just send the output of data in chunks.
- **Concurrent request handling with Asynchronous event-driven IO:** All nodes of API in Node.js are asynchronous, which helps a node to receive a request for an operation. It works in the background, along with taking new requests. Hence it handles all requests concurrently and does not wait for previous responses.

Q2) What is REPL in Node.js?

REPL stands for Reading Eval Print and Loop. Using these operations, you can write programs to accept commands, evaluate those and print them. It supports an

environment similar to Linux or UNIX where a developer can enter commands and get a response with the output. REPL performs the following functions:

- **READ:** It reads input from the user, parses it into JavaScript and then proceeds to store it in the memory.
- **EVAL:** It executes the data structure which stored the information.
- **PRINT:** It prints the outcome which is received from executing the command.
- **LOOP:** It loops the above command until the developer presses Ctrl + C two times.

Q3) What is Callback Hell?

Callback hell is nested callbacks that callback a procedure many times and hence make the code unreadable.

```
downloadPhoto('http://coolcats.com/cat.gif', displayPhoto)
function displayPhoto(error, photo) {
  if (error) console.error('Download error!', error)
  else console.log('Download finished', photo)
}
console.log('Download started')
```

Node.js here first declares the 'display photo' function and then calls the 'downloadPhoto' function and passes displayPhoto as its callback.

Q4) What is Tracing?

This is the basic Node.js Interview Questions that are asked in an interview. Tracing enables you to trace information generated by V8. It can be enabled by passing flag as — trace-events-enabled while starting the node. All these categories that are recorded can be specified by the flag –trace-event-categories. The logs that are enabled can be opened as chrome://tracing in Chrome.

Q5) How to avoid Callback Hell?

Node.js uses only a single thread, and hence this may lead to many queued events. Hence whenever a long-running query finishes its execution, it runs the callback associated with the query. To solve this issue, the following can be followed:

- **Modular code:** This code will be split into smaller modules and later can be joined together to the main module to achieve the desired result.
- **Promise Mechanism:** This is an alternate way for an async code. This mechanism ensures either a result of an error. They take two optional arguments and depending on a state of promise one of them will be called.
- **Use of Generators:** These are routines that wait and resume using the yield keyword. They can also suspend and resume asynchronous operations.
- **Async Mechanism:** This method provides a sequential flow of execution. This module has <async.waterfall> API, which passes data from one operation to another using the next callback. The caller is the main method, and it is called only once through a callback.

Q6) How to load HTML in Node.js?

In order to load HTML in Node.js we should change 'Content-type' in HTML code from plain text to HTML text.

Let us see an example where a static file is created in server:

```
fs.readFile(filename, "binary", function (err, file) {  
  if (err) {  
    response.writeHead(500, { "Content-Type": "text/plain" });  
    response.write(err + "\n");  
    response.end();  
    return;  
  }  
  response.writeHead(200);  
  response.write(file, "binary");  
  response.end();  
});
```

This code can be modified to load as HTML page instead of plain text.

```
fs.readFile(filename, "binary", function (err, file) {  
  if (err) {  
    response.writeHead(500, { "Content-Type": "text/html" });  
    response.write(err + "\n");  
    response.end();  
    return;  
  }  
  response.writeHead(200, { "Content-Type": "text/html" });  
  response.write(file);  
  response.end();  
});
```

Q7) Explain EventEmitter in Node.js?

This is one of the most popular Node.js Interview Questions. The event module in Node.js can have an EventEmitter class which is helpful in raising and handling custom events. It can be accessed by the below code:

```
// Import events module
var events = require('events');
// Create an EventEmitter object
var EventEmitter = new events.EventEmitter();
```

When an error occurs, it also calls the error event. When a new listener is added newListener event is triggered, and similarly when a listener is removed then removeListener is called.

Q8) What is NPM?

NPM stands for Node Package Manager. It has two main functions:

It works on Online Repository for node.js packages which are present at <nodejs.org>.

It works as a command-line utility and does version management.

You can verify the version using the below command:

```
npm -version
```

To install any module, you can use

```
npm install < Module Name >
```

Q9) Explain the use of method spawn() and fork()?

This method is used when a new process is to be launched with a given set of commands. The below command can be used for this purpose:

```
child_process.spawn(command[, args][, options])
```

The fork method is considered to be a special case for the spawn() method. It can be used as below:

```
child_process.fork(modulePath[, args][, options])
```

Q10) Explain the control flow function and steps to execute it?

It is the code that runs between asynchronous function calls. To execute it following steps should be followed:

- Control the order of execution.
- Collect data.
- Limit concurrency.
- Call the next step in the program.

