

Top 50 Node.js Interview Questions and Answers for 2023

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

Factor	Node.js	JavaScript
Engine	V8 – Google Chrome	V8, Spider Monkey, and JS Core
Usage	To perform non-blocking activities	For general client-side operations
Working	Interpreter – Scripting	Programming language

2. What is Node.js?

Node.js is a very popular scripting language that is primarily used for server-side scripting requirements. It has numerous benefits compared to other server-side programming languages out there, the most noteworthy one being the non-blocking I/O.

3. Briefly explain the working of Node.js.

Node.js is an entity that runs in a virtual environment, using JavaScript as the primary scripting language. It uses a simple V8 environment to run on, which helps in the provision of features like the non-blocking I/O and a single-threaded event loop.

4. Where is Node.js used?

Node.js is used in a variety of domains. But, it is very well regarded in the design of the following concepts:

- Network application
- Distributed computing
- Responsive web apps
- Server-Client applications

5. What is the difference between Node.js and Angular?

Node.js	Angular
Used in situations where scalability is a requirement	Best fit for the development of real-time applications
Ability to generate queries in a database	Ability to simplify an application into the MVC architecture

Mainly used to develop small/medium-sized applications	Mainly used to develop real-time interactive web applications
Provides many frameworks such as Sails, Partial, and Express	Angular is an all-in-one web app framework
Coded using C++ and JavaScript	Coded in TypeScript

6. Why is Node.js single-threaded?

Node.js works on the single-threaded model to ensure that there is support for asynchronous processing. With this, it makes it scalable and efficient for applications to provide high performance and efficiency under high amounts of load.

7. What are the different API functions supported by Node.js?

There are two types of API functions. They are as follows:

Synchronous APIs: Used for blocking functions

Asynchronous APIs: Used for non-blocking functions

8. What is the difference between synchronous and asynchronous functions?

Synchronous functions are mainly used for I/O operations. They are instantaneous in providing a response to the data movement on the server and keeping up with the data as per the requirements. If there are no responses, the API will throw an error.

On the other hand, asynchronous functions, as the name suggests, work on the basis of not being synchronous. Here, HTTP requests, when pushed, will not wait for a response to begin. Responses to any previous requests will be continuous, even if the server has already gotten the response.

Next among the Node JS questions, you have to learn about the control flow function.

9. What is the control flow function?

The control flow function is a common code snippet, which executes whenever there are any asynchronous function calls made. It is used to evaluate the order in which these functions are executed in Node.js.

10. Why is Node.js so popular these days?

Node.js has gained an immense amount of traction as it mainly uses JavaScript. It provides programmers with the following options:

- Writing JavaScript on the server
- Access to the HTTP stack
- File I/O entities
- TCP and other protocols
- Direct database access

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

When running an application, callbacks are entities that have to be handled. In the case of Node.js, event loops are used for this purpose. Since Node.js supports non-blocking sending, this is a very important feature to have.

The working of an event loop begins with the occurrence of a callback wherever an event begins. This is usually run by a specific listener. Node.js will keep executing the code after the functions have been called, without expecting the output prior to the beginning.

Once all of the code is executed, outputs are obtained, and the callback function is executed. This works in the form of a continuous loop, hence the name event loop.

12. What are the asynchronous tasks that should occur in an event loop?

Following are some of the tasks that can be done using an event loop asynchronously:

- Blocking send requests
- High computational requirement
- Real-time I/O operations

13. What is the order of execution in control flow statements?

The following is the order in which control flow statements are used to process function calls:

- Handling execution and queue
- Data collection and storage
- Concurrency handling and limiting
- Execution of the next piece of code

14. What are the input arguments for an asynchronous queue?

There are two main arguments that an asynchronous queue uses. They are as follows:

- Concurrency value
- Task function

15. Are there any disadvantages to using Node.js?

A multi-threaded platform can run more effectively and provide better responsiveness when it comes to the execution of intensive CPU computation, and the usage of relational databases with Node.js is already becoming obsolete.

16. What is the primary reason to use the event-based model in Node.js?

The event-based model in Node.js is used to overcome the problems that occur when using blocking operations in the I/O channel.

Next in this blog comprising Node.js questions, you need to understand how you can import libraries into Node.js.

17. How can you import external libraries into Node.js?

External [libraries](#) can be easily imported into Node.js using the following command:

```
var http=require ("http")
```

This command will ensure that the HTTP library is loaded completely, along with the exported object.

Next, among the Node JS questions you need to know about event-driven programming.

18. What is meant by event-driven programming in Node.js?

Event-driven programming is a technique in which the workflow execution of a program is mainly controlled by the occurrence of events from external programs or other sources.

The event-driven architecture consists of two entities, namely:

- Event handling

- Event selection

19. What is the difference between Ajax and Node.js?

Ajax	Node.js
Client-side programming technology	Server-side scripting language
Executes in the browser	Executes on the server

20. What is the framework that is used most often in Node.js today?

Node.js has multiple [frameworks](#), namely:

- Hapi.js
- [Express.js](#)
- Sails.js
- Meteor.js
- Derby.js
- Adonis.js

Among these, the most used framework is Express.js for its ability to provide good scalability, flexibility, and minimalism.

21. What are the security implementations that are present in Node.js?

The following are the important implementations for security:

- Error handling protocols
- Authentication pipelines

22. What is the meaning of a test pyramid?

A test pyramid is a methodology that is used to denote the number of test cases executed in unit testing, integration testing, and combined testing (in that order). This is maintained to ensure that an ample number of test cases are executed for the end-to-end development of a project.

23. What is Libuv?

Libuv is a widely used library present in Node.js. It is used to complement the asynchronous I/O functionality of Node.js. It was developed in-house and used alongside systems such as Luvit, Julia, and more.

The following are some of the features of Libuv:

- [File system](#) event handling
- Child forking and handling
- Asynchronous UDP and TCP sockets
- Asynchronous file handling and operations

Next in these Node JS questions, you need to understand the functioning of Google Chrome.

To learn full-stack development in detail, sign up for this industry-based [Full Stack Web Development Course](#).

24. Why does Google use the V8 engine for Node.js?

Google makes use of the V8 engine because it can easily convert JavaScript into a low-level language. This is done to provide high performance during the execution of an application and also to provide users with real-time abilities to work with the application.

25. What is the difference between spawn and fork methods in Node.js?

The **spawn()** function is used to create a new process and launch it using the command line. It creates a node module on the processor. Node.js invokes this method when the child processes return data.

The following is the syntax for the spawn() method:

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

Coming to the fork() method, it can be considered as an instance of the already existing spawn() method. Spawning ensures that there is more than one active worker node to handle tasks at any given point in time.

The following is the syntax for the fork() method:

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

If you are looking forward to becoming proficient in Angular.js, then make sure to check out Intellipaat's latest offerings for the [Angular JS Course](#).

26. What is the use of middleware in Node.js?

A middleware is a simple function that has the ability to handle incoming requests and outbound response objects. Middleware is used primarily for the following tasks:

- Execution of code (of any type)
- Updating request and response objects
- Completion of request-response iterations
- Calling the next middleware

27. What are global objects in Node.js?

Global objects are objects with a scope that is accessible across all of the modules of the Node.js application. There will not be any need to include the objects in every module. One of the objects is declared as global. So, this is done to provide any functions, strings, or objects access across the application.

Next among the Node JS coding questions, you need to take a look at the usage of assets in Node JS.

28. Why is assert used in Node.js?

Assert is used to explicitly write test cases to verify the working of a piece of code. The following code snippet denotes the usage of assert:

```
var assert = require('assert');
function add(x, y) {
  return x + y;
}
var result = add(3,5);
assert( result === 8, 'three summed with five is eight');
```

29. What are stubs in Node.js?

Stubs are simply functions that are used to assess and analyze individual component behavior. When running test cases, stubs are useful in providing the details of the functions executed.

30. How is a test pyramid implemented using the HTML API in Node.js?

Test pyramids are implemented by defining the HTML API. This is done using the following:

- A higher number of unit test cases
- A smaller number of integration test methods
- A fewer number of HTTP endpoint test cases

31. Why is a buffer class used in Node.js?

A [buffer class](#) is primarily used as a way to store data in Node.js. This can be considered as a similar implementation of arrays or lists. Here, the class refers to a raw memory location that is not present in the V8 heap structure.

The buffer class is global, thereby extending its usage across all the modules of an application.

32. Why is Express.js used?

[Node JS Express](#) is a widely used framework built using Node.js. Express.js uses a management point that controls data flow between servers and server-side applications.

Being lightweight and flexible, Express.js provides users with lots of features used to design mobile applications.

33. What is the use of the connect module in Node.js?

The connect [module in Node.js](#) is used to provide communication between Node.js and the HTTP module. This also provides easy integration with Express.js, using the middleware modules.

34. What are streams in Node.js?

Streams are a set of data entities in Node.js. These can be considered similar to the working of strings and array objects. Streams are used for continuous read/write operations across a channel. But, if the channel is unavailable, all the data cannot be pushed to the memory at once. Hence, using streams will make it easy to process a large set of data in a continuous manner.

Next up on this compilation of top Node.js interview questions for experienced, let's check out the advanced category of questions.

35. What are the types of streams available in Node.js?

Node.js supports a variety of streams, namely:

- Duplex (both read and write)
- Readable streams
- Writable streams
- Transform (duplex for modifying data)

36. What is the use of REPL in Node.js?

REPL stands for Read-Eval-Print-Loop. It provides users with a virtual environment to test JavaScript code in Node.js.

To launch REPL, a simple command called 'node' is used. After this, JavaScript commands can be typed directly into the command line.

37. What is meant by tracing in Node.js?

Tracing is a methodology used to collect all of the tracing information that gets generated by V8, the node core, and the userspace code. All of these are dumped into a log file and are very useful to validate and check the integrity of the information being passed.

38. Where is package.json used in Node.js?

The 'package.json' file is a file that contains the metadata about all items in a project. It can also be used as a project identifier and deployed as a means to handle all of the project dependencies.

39. What is the difference between readFile and createReadStream in Node.js?

- **readFile:** This is used to read all of the contents of a given file in an asynchronous manner. All of the content will be read into the memory before users can access it.
- **create ReadStream:** This is used to break up the field into smaller chunks and then read it. The default chunk size is 64 KB, and this can be changed as per requirement.

40. What is the purpose of the crypto module in Node.js?

The crypto module in Node.js is used to provide users with cryptographic functionalities. This provides them with a large number of wrappers to perform various operations such as ciphering, deciphering, signing, and hashing.

41. What is a passport in Node.js?

Passport is a widely used middleware present in Node.js. It is primarily used for authentication, and it can easily fit into any Express.js-based web application.

With every application created, it will require unique authentication mechanisms. This is provided as single modules by using a passport, and it becomes easy to assign strategies to applications based on requirements, thereby avoiding any sort of dependencies.

42. How to get information about a file in Node.js?

The `fs.stat` function is used to get the required information from a file.

The syntax is as follows:

```
fs.stat(path, callback)
```

Here,

- Path: The string that has the path to the name.
- Callback: The callback function where stats is an object of `fs.stats`

43. How does the DNS lookup function work in Node.js?

The DNS lookup method uses a web address as its parameter and returns the IPv4 or IPv6 record, correspondingly.

There are other parameters, such as the options used to set the input as an integer or an object. If nothing is provided here, both IPv4 and IPv6 are considered. The third parameter is for the callback function.

The syntax is as follows:

```
dns.lookup(address, options, callback)
```

44. What is the use of EventEmitter in Node.js?

Every single object in Node.js that emits is nothing but an instance of the `EventEmitter` class. These objects have a function that is used to allow the attachment between the objects and the named events.

Synchronous attachments of the functions are done when the `EventEmitter` object emits an event.

45. What is the difference between `setImmediate()` and `setTimeout()`?

The `setImmediate()` function is meant to execute a single script once the current event loop is complete.

The `setTimeout()` function is used to hold a script and schedule it to be run after a certain time threshold is reached.

The order of execution will solely depend on the context in which the functions are called. If called from the main module, the timing will be based on the performance of the process.

46. What is the use of `module.exports` in Node.js?

The `module.exports` function is used to expose two functions and bring them into a usable context. A module is an entity used to store relative code in a single snippet. This can be considered an operation to move all of the functions into one single file.