Node.js Interview Questions

edureka!

edureka!

© Brain4ce Education Solutions Pvt. Ltd.

1. What is Node.js?

Node.js is an extremely powerful framework developed on **Chrome's V8 JavaScript engine** that compiles the JavaScript directly into the native machine code. It is a lightweight framework used for creating server-side web applications and extends JavaScript API to offer usual server-side functionalities. It is generally used for large-scale application development, especially for video streaming sites, single-page applications, and other web applications.

2. What is the advantage of using node.js?

- It provides an easy way to build scalable network programs
- Generally fast
- Great concurrency
- Asynchronous everything
- Almost never blocks

3. Why is Node.js single threaded?

Node.js uses a single-threaded model to support async processing. With async processing, an application can perform better and is more scalable under web loads. Thus, Node.js makes use of a single-threaded model approach rather than a typical thread-based implementation.

4. What is NPM? Explain the need for NPM.

Node Package Manager (NPM) is the default package manager of Node.js that is completely written in JavaScript. It is responsible for managing all the Node.js packages and modules.

The main two functionalities of NPM are:

- It provides online repositories for packages/modules for Node.js, which you can easily search online on their official site.

- It also provides a Command Line Interface (CLI), which helps the developers in locally interacting with their systems.

Some of the important purpose of NPM:

- It helps in incorporating the pre-built packages into our project
- It assists in downloading various standalone tools which can be used right away
- Using npx, you can even run packages without having to download it
- Developers often use NPM to share their code with other NPM users across the globe

5. What is the need to install a package globally?

When a package is installed globally, it can be accessed by any other project without installing it separately. Whereas when a package is installed locally, it can be used only by the project where it has been installed. Globally installed packages are stored in <user-directory>/npm directory.

6. What is the purpose of a package.json file?

The package.json file contains the metadata of a project where the properties of a package are defined. It acts as the heart of an application.

7. Name some of the widely used Node.js libraries?

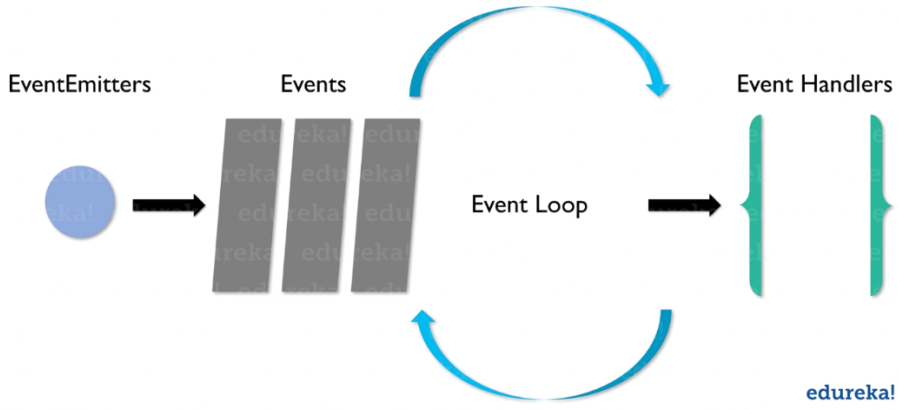
Two of the most commonly used Node.js libraries are:

- **ExpressJS** - It is a web application framework that provides a wide set of features that make web application development easier and faster.

- **Mongoose** - It is an object modeling tool that is used to create a connection from the application to the database.

8. What is an Event loop in Node.js, and how does it work?

An **Event loop** in Node.js handles all the asynchronous callbacks in an application. It is one of the most important aspects of Node.js, and the reason behind Node.js have non-blocking I/O. Since Node.js is an event-driven language, you can easily attach a listener to an event, and then when the event occurs, the callback will be executed by the specific listener. Whenever functions like setTimeout, http.get, and fs.readFile is called, Node.js executed the event loop and then proceeds with the further code without waiting for the output. Once the entire operation is finished, Node.js receives the output and then executes the callback function. Therefore, all the callback functions are placed in a queue in a loop. Once the response is received, they are executed one by one.



9. Explain the purpose of module.exports?

A module in Node.js is used to encapsulate all the related codes into a single unit of code which can be interpreted by shifting all related functions into a single file.

10. Explain the concept of middleware in Node.js?

In general, middleware is a function that receives the Request and Response objects. In other words, in an application's request-response cycle, these functions have access to various request & response objects along with the next function of the cycle. The next function of middleware is represented with the help of a variable, usually named next.

Most performed tasks by the middleware functions are:

- Execute any type of code
- Update or modify the request and the response objects
- Finish the request-response cycle
- Invoke the next middleware in the stack

11. For Node.js, why Google uses the V8 engine?

Google uses **V8** as it is a Chrome runtime engine that converts JavaScript code into native machine code. This, in turn, speeds up the application execution and response process and gives you a fast-running application.