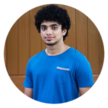
20 Frequently Asked Node.js Interview Questions and Answers

Srihari Sasikumar



Last updated October 11, 2018

196504 Views

Node.js is a server-side technology that you have to be proficient in to become the ideal MEAN stack developer. It is one of the most rewarding web development tools to learn, not only because of the lucrative salary—averaging $100,000/ year—but also the fast and continued growth of job postings requiring knowledge of the technology.

If you have a MEAN stack developer interview lined up or are anticipating interviews soon, expect to face cutthroat competition. The 20 interview questions outlined below should help you prepare adequately to answer every question asked confidently.

Node.js is a server side scripting based on Google's V8 JavaScript engine. It is used to build scalable programs especially web applications that are computationally simple but are frequently accessed.

You can use Node.js in developing I/O intensive web applications like video streaming sites. You can also use it for developing: Real-time web applications, Network applications, General-purpose applications and Distributed systems.

2. Why use Node.js?

Node.js makes building scalable network programs easy. Some of its advantages include:

- It is generally fast
- It almost never blocks
- It offers a unified programming language and data type
- Everything is asynchronous
- It yields great concurrency

3. What are the features of Node.js?

Node.js is a single-threaded but highly scalable system that utilizes JavaScript as its scripting language. It uses asynchronous, event-driven I/O instead of separate processes or threads. It is able to achieve high output via single-threaded event loop and non-blocking I/O.

4. How else can the JavaScript code below be written using Node.Js to produce the same output?

```
}, 0);
console.log("third");
```

Output:

```
first
third
second
```

In Node.js version 0.10 or higher, setImmediate(fn) will be used in place of setTimeout(fn,0) since it is faster. As such, the code can be written as follows:

```
console.log("first");
setImmediate(function(){
   console.log("second");
});
console.log("third");
```

5. How do you update NPM to a new version in Node.js?

You use the following commands to update NPM to a new version:

```
$ sudo npm install npm -g
/usr/bin/npm -> /usr/lib/node_modules/npm/bin/npm-cli.js
npm@2.7.1 /usr/lib/node_modules/npm
```

6. Why is Node.js Single-threaded?

Node.js is single-threaded for async processing. By doing async processing on a single-thread under typical web loads, more performance and scalability can be achieved as opposed to the typical thread-based implementation.

A callback function is called at the completion of a given task. This allows other code to be run in the meantime and prevents any blocking. Being an asynchronous platform, Node.js heavily relies on callback. All APIs of Node are written to support callbacks.

8. What is callback hell in Node.js?

Callback hell is the result of heavily nested callbacks that make the code not only unreadable but also difficult to maintain. For example:

```
query(″SELECT clientId FROM clients WHERE clientName=′picanteverde′;″, function(id){
  query(″SELECT * FROM transactions WHERE clientId=″ + id, function(transactions){
    transactions.each(function(transac){
         query(″UPDATE transactions SET value = ″ + (transac.value*0.1) + ″ WHERE id=″ + transac.id, function(error){
        if(!error){
          console.log(″success!!″);
        }else{
          console.log(″error″);
        }
      });
    });
  });
});
```

9. How do you prevent/fix callback hell?

The three ways to prevent/fix callback hell are:

- Handle every single error
- Keep your code shallow
- Modularize – split the callbacks into smaller, independent functions that can be called with some parameters then joining them to achieve desired results.

The first level of improving the code above might be:

```
      console.log("success!!");
    }else{
      console.log("error");
    }
  },
  updateTransaction = function(t){
     query("UPDATE transactions SET value = " + (t.value*0.1) + " WHERE id=" + t.id, logError);
  },
  handleTransactions = function(transactions){
    transactions.each(updateTransaction);
  },
  handleClient = function(id){
    query("SELECT * FROM transactions WHERE clientId=" + id, handleTransactions);
  };
```

```
query("SELECT clientId FROM clients WHERE clientName='picanteverde';",handleClient);
```

You can also use Promises, Generators and Async functions to fix callback hell.

10. Explain the role of REPL in Node.js.

As the name suggests, REPL (Read Eval print Loop) performs the tasks of – Read, Evaluate, Print and Loop. The REPL in Node.js is used to execute ad-hoc Javascript statements. The REPL shell allows entry to javascript directly into a shell prompt and evaluates the results. For the purpose of testing, debugging, or experimenting, REPL is very critical.

11. Name the types of API functions in Node.js.

There are two types of functions in Node.js.:

- Blocking functions - In a blocking operation, all other code is blocked from executing until an I/O event that is being waited on occurs. Blocking functions execute synchronously

```
console.log(data);
// moreWork(); will run after console.log
```

The second line of code blocks the execution of additional JavaScript until the entire file is read. moreWork () will only be called after Console.log

- Non-blocking functions - In a non-blocking operation, multiple I/O calls can be performed without the execution of the program being halted. Non-blocking functions execute asynchronously.

For example:

```
const fs = require('fs');
fs.readFile('/file.md', (err, data) => {
  if (err) throw err;
  console.log(data);
});
// moreWork(); will run before console.log
```

Since fs.readFile () is non-blocking, moreWork () does not have to wait for the file read to complete before being called. This allows for higher throughput.

12. Which is the first argument typically passed to a Node.js callback handler?

Typically, the first argument to any callback handler is an optional error object. The argument is null or undefined if there is no error.

Error handling by a typical callback handler could be as follows:

```
function callback(err, results) {
    // usually we'll check for the error before handling results
    if(err) {
        // handle error somehow and return
    }
    // no error, perform standard callback handling
}
```

13. What are the functionalities of NPM in Node.js?

NPM (Node package Manager) provides two functionalities:

- Online repository for Node.js packages
- Command line utility for installing packages, version management and dependency management of Node.js packages

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

Node.js and Ajax (Asynchronous JavaScript and XML) are the advanced implementation of JavaScript. They all serve completely different purposes.

Ajax is primarily designed for dynamically updating a particular section of a page's content, without having to update the entire page.

Node.js is used for developing client-server applications.

15. Explain chaining in Node.js.

Chaining is a mechanism whereby the output of one stream is connected to another stream creating a chain of multiple stream operations.

16. What are "streams" in Node.js? Explain the different types of streams present in Node.js.

Streams are objects that allow reading of data from the source and writing of data to the destination as a continuous process.

There are four types of streams.

- to facilitate both read and write operations
- is a form of Duplex stream that performs computations based on the available input

17. What are exit codes in Node.js? List some exit codes.

Exit codes are specific codes that are used to end a “process” (a global object used to represent a node process).

Examples of exit codes include:

- Unused
- Uncaught Fatal Exception
- Fatal Error
- Non-function Internal Exception Handler
- Internal Exception handler Run-Time Failure
- Internal JavaScript Evaluation Failure

18. What are Globals in Node.js?

Three keywords in Node.js constitute as Globals. These are:

- **Global** – it represents the Global namespace object and acts as a container for all other objects.
- **Process** – It is one of the global objects but can turn a synchronous function into an async callback. It can be accessed from anywhere in the code and it primarily gives back information about the application or the environment.

Other Segments

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

Angular.JS is a web application development framework while Node.js is a runtime system.

20. Why is consistent style important and what tools can be used to assure it?

Consistent style helps team members modify projects easily without having to get used to a new style every time. Tools that can help include Standard and ESLint.

Conclusion

Since every question has more than one answer, feel free to personalize your answers as much as possible, especially when you have work experience to relate the answers to. In case you need a refresher in Node.js or if you do not have any training in it, you should consider Simplilearn’s Node.js certification training. This course provides an in-depth knowledge of concepts such as shrink-wrap, asynchronous programming, Node Packet Manager (NPM) and more.

About the Author

Srihari Sasikumar is a Product Manager with over six years of experience in various industries including Information Technology, E-Commerce, and E-Learning. Srihari follows the key trends in Big Data, Data Science, Programming & AI very closely.

Recommended articles for you

Node.js for Beginners: How to Get Started

Everything You Need to Know for a Hadoop Developer Interview

Other Segments

VMware vSphere Best Practices

LEAVE COMMENT

1 Comment Simplilearn 1 Login

Recommend 9 Tweet Share Sort by Best

Join the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

Tarun Nagpal • a year ago

Good set of questions. Can you guide me, where i can find more ?

• Reply • Share ›

Subscribe Add Disqus to your siteAdd DisqusAdd Disqus' Privacy PolicyPrivacy PolicyPrivacy

SAP Modules – SAP FI, SAP CO, SAP SD, SAP HCM and more

Other Segments

Eshna Verma

Published on Oct 15, 2012

527067 Views 140 Comments

Overview of SAP Modules

SAP Certification exam is based on Enterprise Resource Planning solutions, typically SAP software solutions. Enterprise Resource Planning (ERP) that allows an enterprise to manage databases for different processes from a single unified system. ERP system was developed initially for inventory control, however over the years; different software was developed for different processes in an enterprise for better data management and work flow. In today's world, there are many leading market providers of ERP system and SAP is one of the market and technology leaders in building business software towards structured work and data management in organizations. SAP AG is originally German multinational software who deals in software development for business management and improved customer relations in enterprises. Systems Analysis and Program Development (SAP) was founded in June, 1972 and since then, many SAP ERP operations modules has emerged that are designed focusing on various different processes including SAP ERP sales and service, sales and distribution, customer relationship, financial management, business intelligence and more. Let's get an overview of few of the SAP development modules in this post.

Other Segments

- SAP Sales and Distribution (SD)
- SAP Production Planning (PP)
- SAP Materials Management (MM)
- SAP Quality Management (QM)
- SAP Human Capital Management (HCM)

SAP Financial Accounting (FI)

SAP FI module as the term suggests deals in managing financial transactions within enterprises. This financial accounting module helps employees to manage data involved in any financial and business transactions in a unified system. This module functions very well for reporting requirements. The SAP FI module is very flexible and functions well in any type of economic situation. Be it a smaller organization or a larger organization, SAP implementation helps in consolidating data for diverse business transactions and legal requirements. Financial Accounting module helps one to get real-

Other Segments

SAP Controlling (CO)

SAP CO module is another important SAP modules offered to enterprises. The controlling module supports in the process works of planning, reporting and monitoring operations of businesses. It involves methods to view and organize costs that are required for financial reporting. Controlling module enables one to plan, track, perform and report about costs. Controlling includes managing and configuring master data that covers cost elements, cost centers, profit centers, internal orders, and functional area and so on.

SAP Sales and Distribution (SD)

SAP SD modules deal in managing all transactions ranging from enquiries, proposals, quotations, pricing and more. The sales and distribution module helps greatly in inventory control and management. SAP SD module consists of master data, system configuration and transactions. Some of the sub-components of SAP SD module are: master data, sales support, sales, shipping and transportation, billing, credit management, sales information system and so on.

SAP Production Planning (PP)

SAP PP module is another important module that includes software designed specifically for production planning and management. This module also consists of master data, system configuration and transactions in order to accomplish plan procedure for production. SAP PP module collaborate with master data, sales and operations planning, distribution resource planning, material requirements planning, Kanban, product cost planning and so on while working towards production management in enterprises.

SAP Materials Management (MM)

SAP MM module as the term suggests manages materials required, processed and produced in enterprises. Different types of procurement processes are managed with this system. Some of the popular sub-components in SAP MM module are vendor master data, consumption based planning, purchasing, inventory management, invoice verification and so on.

SAP Quality Management (QM)

s. SAP QM module collaborates in procurement and sales, production, planning, inspection , notification, control, audit management and so on.

SAP Human Capital Management (HCM)

SAP HCM module enhances the work process and data management within HR department of enterprises. Right from hiring a person to evaluating one's performance, managing promotions, compensations, handling payroll and other related activities of an HR is processed using this module. The task of managing the details and task flow of the most important resource i.e. human resource is managed using this SAP ERP HCM module.

These are some of the most important SAP deployment modules that develop software for efficient work in enterprises.

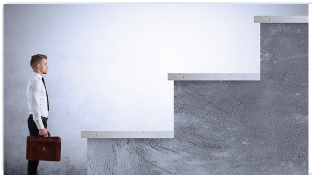
About the Author

Eshna writes on PMP, PRINCE2, ITIL, ITSM, & Ethical Hacking. She has done her Masters in Journalism and Mass Communication and is a Gold Medalist in the same. A voracious reader, she has penned several articles in leading national newspapers like TOI, HT, and The Telegraph. She loves travelling and photography.

Recommended articles for you

SAP Financials and SAP Accounting Modules

SAP Sales and Distribution, SAP SD Configuration and S...



Starting a Career as a SAP FICO Consultant

Other Segments ▾