

Node.js Interview Questions And Answers Guide.



Global Guideline.

<https://globalguideline.com/>



Node.js Job Interview Preparation Guide.

Question # 1

What tools and IDEs are used for Node.js?

Answer:-

IDEs (editor, debugger, linter)

- * Atom (free open-source)
- * Nodeclipse Enide Studio (free open-source, Eclipse-based)
- * JetBrains WebStorm (commercial)
- * JetBrains IntelliJ IDEA (commercial)
- * Microsoft Visual Studio with TypeScript
- * NoFlo "flow-based programming environment integrated with GNOME APIs"

[Read More Answers.](#)

Question # 2

How to get started with Node.js?

Answer:-

First, learn the core concepts of Node.js:

You'll want to understand the asynchronous coding style that Node encourages.

Async != concurrent. Understand Node's event loop!

Node uses CommonJS-style require() for code loading; it's probably a bit different from what you're used to.

Familiarize yourself with Node's standard library.

Then, you're going to want to see what the community has to offer:

The gold standard for Node package management is NPM.

It is a command line tool for managing your project's dependencies.

Make sure you understand how Node and NPM interact with your project via the node_modules folder and package.json.

NPM is also a registry of pretty much every Node package out there

Finally, you're going to want to know what some of the more popular packages are for various tasks:

Useful Tools for Every Project:

Underscore contains just about every core utility method you want.

CoffeeScript makes JavaScript considerably more bearable, while also keeping you out of trouble!

Caveat: A large portion of the community frowns upon it. If you are writing a library, you should consider regular JavaScript, to benefit from wider collaboration.

Unit Testing:

Mocha is a popular test framework.

Vows is a fantastic take on asynchronous testing, albeit somewhat stale.

Expresso is a more traditional unit testing framework.

node-unit is another relatively traditional unit testing framework.

Web Frameworks:

Express is by far the most popular framework.

Meteor bundles together jQuery, Handlebars, Node.js, websockets, MongoDB, and DDP and promotes convention over configuration without being a Rails clone.

Tower is an abstraction of top of Express that aims to be a Rails clone.

Geddy is another take on web frameworks.

RailwayJS is a Ruby-on-Rails inspired MVC web framework.

SailsJS is a realtime MVC web framework.

Sleek.js is a simple web framework, built upon express.js.

Hapi is a configuration-centric framework with built-in support for input validation, caching, authentication, etc.

Koa Koa is a new web framework designed by the team behind Express, which aims to be a smaller, more expressive, and more robust foundation for web applications and APIs.

Web Framework Tools:

Jade is the HAML/Slim of the Node world

EJS is a more traditional templating language.

Don't forget about Underscore's template method!

Networking:

Connect is the Rack or WSGI of the Node world.

Request is a very popular HTTP request library.

socket.io is handy for building WebSocket servers.

Command Line Interaction:



Node.js Interview Questions And Answers

Optimist makes argument parsing a joy.
Commander is another popular argument parser.
Colors makes your CLI output pretty.

[Read More Answers.](#)

Question # 3

Can we use jQuery with Node.js?

Answer:-

No. It's going to be quite a big effort to port a browser environment to node.

Another approach, that I'm currently investigating for unit testing, is to create "Mock" version of jQuery that provides callbacks whenever a selector is called. This way you could unit test your jQuery plugins without actually having a DOM. You'll still have to test in real browsers to see if your code works in the wild, but if you discover browser specific issues, you can easily "mock" those in your unit tests as well. I'll push something to github.com/felixge once it's ready to show.

[Read More Answers.](#)

Question # 4

Do you know what mean stack means?

Answer:-

MEAN stack stands for MongoDB-Express-AngularJS-NodeJS.

First of all, MEAN is JS-only environment. Express is one of the most popular web app frameworks for NodeJS. NodeJS development often goes along noSQL database, like CouchDB or MongoDB, due to their document based structure and JSON notation, which is suitable for all-JS nature of NodeJS. AngularJS is a lightweight framework for building client-side applications.

[Read More Answers.](#)

Question # 5

What is Node.js?

Answer:-

Node.js is a software platform for scalable server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on Mac OS X, Windows and Linux with no changes.

[Read More Answers.](#)

Question # 6

How to extract POST data in node.js?

Answer:-

If you use Express (High performance, high class web development for Node.js), you can do this:

HTML:

```
<form method="post" action="/">
  <input type="text" name="user[name]">
  <input type="text" name="user[email]">
  <input type="submit" value="Submit">
</form>
```

Javascript:

```
app.use(express.bodyParser());
app.post('/', function(request, response){
  console.log(request.body.user.name);
  console.log(request.body.user.email);
});
```

[Read More Answers.](#)

Question # 7

Explain Node.js Architecture?

Answer:-

There are four building blocks that constitute Node.js. First, Node.js encapsulates libuv to handle asynchronous events and Google's V8 to provide a run-time for JavaScript. Libuv is what abstracts away all of the underlying network and file system functionality on both Windows and POSIX-based systems like Linux, Mac OS X and Unix. The core functionality of Node.js, modules like Assert, HTTP, Crypto, etc., reside in a core library written in JavaScript. The Node.js bindings, written in C++, provide the glue connecting these technologies to each other and to the operating system.

[Read More Answers.](#)

Question # 8

Tell me how to decide when to use Node.js?

Answer:-

You did a great job of summarizing what's awesome about Node.js. My feeling is that Node.js is especially suited for applications where you'd like to maintain a persistent connection from the browser back to the server. Using a technique known as "long-polling", you can write an application that sends updates to the user in real time. Doing long polling on many of the web's giants, like Ruby on Rails or Django, would create immense load on the server, because each active client eats up one server process. This situation amounts to a tarpit attack. When you use something like Node.js, the server has no need of maintaining separate threads for each open connection.

[Read More Answers.](#)



Question # 9

Do you know how to get POST query in express node.js?

Answer:-

Things have changed in express 3.0.

Firstly you need to add some middleware to parse the post data of the body.

Add one or both of the following lines of code:

```
app.use(express.json()); // to support JSON-encoded bodies
app.use(express.urlencoded()); // to support URL-encoded bodies
Then, in your handler, use the req.body object:
// assuming POST: name=foo&color=red <-- URL encoding
//
// or POST: {"name":"foo","color":"red"} <-- JSON encoding
app.post('/test-page', function(req, res) {
  var name = req.body.name,
      color = req.body.color;
  // ...
});
```

[Read More Answers.](#)

Question # 10

Do you know how to debug node.js applications?

Answer:-

There's a few tools and ways out there:

Interactive Stack Traces with traceGL - Shareware

Guide here

Profiling with Profiler

Install globally npm install -g profiler

Start your process with node --prof this will create a v8.log file

Build nprof by running ~/.nvm/v0.8.22/lib/node_modules/profiler/tools/build-nprof

Run ~/.nvm/v0.8.22/lib/node_modules/profiler/nprof this will read the v8.log profile and give you nice output

CPU and Memory Profiling with NodeTime

Install to your app npm install nodetime

Include in your app require('nodetime').profile()

Follow the instructions it will output to console

Alternatively, you may want to use look, which is based on nodetime but doesn't send data to nodetime.com.

Blink (formerly Webkit) Developer Tools Debugging with Node Inspector

Install it globally: npm install -g node-inspector

Run your app in debug mode: node --debug-brk your/node/program.js (or attach to a running process: kill -s USR1 <your node process id>)

In another terminal window run node-inspector: node-inspector

Open http://127.0.0.1:8080/debug?port=5858 (or debug remotely by replacing 127.0.0.1 with your host; make sure port 8080 is open)

Webkit Developer Tools Profiling with Node Webkit Agent

Install to your app npm install webkit-devtools-agent

Include in your app agent = require('webkit-devtools-agent')

Activate the agent: kill -SIGUSR2 <your node process id>

Access the agent via the appropriate link

Interactive Cloud9 Debugging

Guide here

Heapdumps to Webkit Developer Tools

Tool and guide here

Logging Libraries that output Debugging Information

Caterpillar

Tracer

Flamegraphs with Dtrace and StackVis

Only supported on SmartOS

Flamegraphs with Chrome Developer Tools

Coming soon

Benchmark

with Apache Bench: ab -n 100000 -c 1 http://127.0.0.1:9778/

with wrk

[Read More Answers.](#)

Question # 11

Tell me what is the purpose of Node.js module.exports and how do you use it?

Answer:-

module.exports is the object that's actually returned as the result of a require call.

The exports variable is initially set to that same object (i.e. it's a shorthand "alias"), so in the module code you would usually write something like this:

```
var myFunc1 = function() { ... };
var myFunc2 = function() { ... };
exports.myFunc1 = myFunc1;
exports.myFunc2 = myFunc2;
```

to export (or "expose") the internally scoped functions myFunc1 and myFunc2.

And in the calling code you would use:

```
var m = require('mymodule');
m.myFunc1();
```

where the last line shows how the result of require is (usually) just a plain object whose properties may be accessed.

[Read More Answers.](#)



Question # 12

Tell me how to make an HTTP POST request in node.js?

Answer:-

This gets a lot easier if you use the request library.

```
var request = require('request');
request.post(
  'http://www.abcsite.com/formpage',
  { form: { key: 'value' } },
  function (error, response, body) {
    if (!error && response.statusCode === 200) {
      console.log(body)
    }
  }
);
```

Aside from providing a nice syntax it makes json requests easy, handles oauth signing (for twitter, etc.), can do multi-part forms (e.g. for uploading files) and streaming.

[Read More Answers.](#)

Question # 13

Are you sure node.js execute system command synchronously?

Answer:-

There's an excellent module for flow control in node.js called asyncblock. If wrapping the code in a function is OK for your case, the following sample may be considered:

```
var asyncblock = require('asyncblock');
var exec = require('child_process').exec;
asyncblock(function (flow) {
  exec('node -v', flow.add());
  result = flow.wait();
  console.log(result); // There'll be trailing
in the output
  // Some other jobs
  console.log('More results like if it were sync...');
});
```

[Read More Answers.](#)

Question # 14

Tell me how to use underscore.js as a template engine?

Answer:-

Everything you need to know about underscore template is here. Only 3 things to keep in mind:

- <% %> - to execute some code
- <%= %> - to print some value in template
- <%- %> - to print some values with HTML escaped

That's all about it.

Simple example:

```
var foo = "blahblah";
var tpl = _.template("<h1>Heading for: <= foo %></h1>");
this would be rendered in string <h1>Heading for: blahblah</h1>
```

[Read More Answers.](#)

Question # 15

Is Node.js on multi-core machines?

Answer:-

Yes, Node.js is one-thread-per-process. This is a very deliberate design decision and eliminates the need to deal with locking semantics. If you don't agree with this, you probably don't yet realize just how insanely hard it is to debug multi-threaded code. For a deeper explanation of the Node.js process model and why it works this way (and why it will NEVER support multiple threads), read my other post.

[Read More Answers.](#)

Computer Programming Most Popular Interview Topics.

- 1 : [PHP Frequently Asked Interview Questions and Answers Guide.](#)
- 2 : [C++ Programming Frequently Asked Interview Questions and Answers Guide.](#)
- 3 : [C Programming Frequently Asked Interview Questions and Answers Guide.](#)
- 4 : [Software engineering Frequently Asked Interview Questions and Answers Guide.](#)
- 5 : [Cobol Frequently Asked Interview Questions and Answers Guide.](#)
- 6 : [Visual Basic \(VB\) Frequently Asked Interview Questions and Answers Guide.](#)
- 7 : [Socket Programming Frequently Asked Interview Questions and Answers Guide.](#)
- 8 : [Perl Programming Frequently Asked Interview Questions and Answers Guide.](#)
- 9 : [VBA Frequently Asked Interview Questions and Answers Guide.](#)
- 10 : [OOP Frequently Asked Interview Questions and Answers Guide.](#)

About Global Guideline.

Global Guideline is a platform to develop your own skills with thousands of job interview questions and web tutorials for fresher's and experienced candidates. These interview questions and web tutorials will help you strengthen your technical skills, prepare for the interviews and quickly revise the concepts. Global Guideline invite you to unlock your potentials with thousands of [Interview Questions with Answers](#) and much more. Learn the most common technologies at Global Guideline. We will help you to explore the resources of the World Wide Web and develop your own skills from the basics to the advanced. Here you will learn anything quite easily and you will really enjoy while learning. Global Guideline will help you to become a professional and Expert, well prepared for the future.

* This PDF was generated from <https://GlobalGuideline.com> at **November 29th, 2023**

* If any answer or question is incorrect or inappropriate or you have correct answer or you found any problem in this document then don't hesitate feel free and [e-mail us](#) we will fix it.

You can follow us on FaceBook for latest Jobs, Updates and other interviews material.
www.facebook.com/InterviewQuestionsAnswers

Follow us on Twitter for latest Jobs and interview preparation guides
<https://twitter.com/InterviewGuide>

Best Of Luck.

Global Guideline Team
<https://GlobalGuideline.com>
Info@globalguideline.com