

Express JS Interview Questions

 mindmajix.com/express-js-interview-questions

Madhuri Yerukala

April 22, 2021

Express JS is a flexible and back-end web application Node JS framework with a powerful set of web and mobile application features. It was released as open-source software licensed under MIT in 2010. We can design single-page, multi-page, and hybrid web applications as well as APIs with the Express JS framework. It is built over Node JS, helping manage routes and servers. And it has been conferred the title of standard server framework for Node JS. It provides several HTTP utility methods to the user for creating APIs and apps easily.



1. How do you install an express application generator for scaffolding?

Express application generator is used for quickly creating an application skeleton. The given command is used for installing the express application generator.

```
npm install expr  
ess-generator -g  
express myApp
```

It will create the project "myApp" with some files. Then we install all the dependencies stated in package.json using the given command:

```
cd myApp  
npm install
```

2. How do you install a yeoman for scaffolding?

Generators are used by yeoman for scaffolding the applications. And we can use the following command to install yeoman.

```
npm install -g yeoman
```

3. Mention the arguments that are available in an Express JS route handler function.

The arguments that are available in the route handler function of Express JS are given below:

- **Res** - It is the response object.
- **Req** - It is the request object
- **Next (optional)** - This argument is used for passing the management to any of the above-given route handlers.

[Learn Complete [***Express JS Tutorial***](#)]

4. Mention the ways of debugging on Linux as well as Windows.

Debugging on Windows can be done as follows:

```
set DEBUG = express:*  
node app.js
```

And debugging on Linux can be done as follows:

```
DEBUG = express:*  
node app.js
```

5. List the built-in middleware functions provided by Express.

Express JS provides the following built-in middleware functions:

1. **Static:** We use it for serving static assets like images, HTML files, etc.
2. **JSON:** This is available in Express 4.16.0+. And we use it for passing the incoming requests with JSON payloads.
3. **URL encoded:** This is also available in Express 4.16.0+. And we use it for passing the incoming requests with URL-encoded payloads.

6. Mention some third-party middleware provided by Express JS.

Some of the many third-party middlewares that the Express JS provides are:

- Cookie-parser
- Body-parser
- Cors
- Mongoose
- Express-validator
- Sequelize.

7. When is application-level Middleware used?

We use the application-level Middleware for binding the app object with the help of the app.use() method. It can be applied on all routes. The syntax is given below:

```
// This Middleware executes for each route.  
App.use(function (req, res, next) {  
  console.log('Current Time:', Date.now())  
  next()  
})
```

8. Tell us about Router-level Middleware and Built-in Middleware.

Router-level Middleware - We use the router-level Middleware for binding with a particular instance of Express.Router().

Built-in Middleware - The version 4.x of Express introduced the built-in Middleware. The dependency on connecting gets removed by use of this Middleware.

9. Mention some of the databases with which Express JS is integrated.

A myriad of NoSQL and RDBMS databases are supported by Express JS, such as:

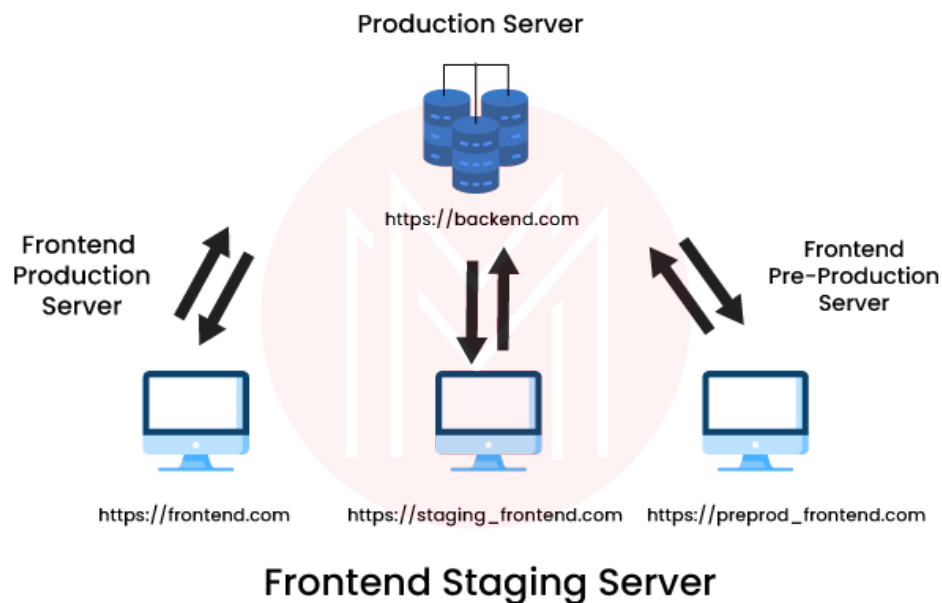
- MySQL
- MongoDB
- PostgreSQL
- SQLite
- Oracle
- SQLite.

Related Article MongoDB vs PostgreSQL

10. What is meant by CORS in Express JS? And what are the ways by which it can be achieved?

CORS is the acronym for Cross-origin resource sharing. We can request the restricted resources from another server or domain using this mechanism. And we can do this mainly in the following three ways:

1. Express cors module
2. Res.header() (or res.set()): Multiple headers can be set using this way.
3. Res.setHeader(): Only a single header can be set in this way.



11. What ways are provided by Express JS to configure the properties?

Express JS provides us with two ways for configuring the properties, which are given below:

- With process.ENV
- With require.JS.

12. How are the properties configured with process.ENV?

The properties are configured by the given steps:

- We create a file within the project folder and name it ".env."
- All other properties are let to be separate within the ".env" file.
- We can employ any of the properties in server.js.

13. How are the properties configured with require.JS?

The properties are configured by the given steps:

- We create a file within the config folder of the project folder and name it "config.json."
- All the config properties are present there within the config.json file.

14. How can the Express JS application be structured?

There is no specific answer to this question. The dimensions of our application and hence the concerned team define the solution in different situations. The express logic in the Routes and alternative applications can board as many files as we want in any directory structure. The given examples can be read for further inspiration:

- Route map
- Route listings
- MVC vogue controllers.

15. How is the plain HTML rendered?

We don't need to render HTML with the function- res.render(). Instead, we can use the res.sendFile() function if we have a specific file. And we can use the Express.static() middleware function if we serve several assets from a directory.

Express JS Interview Questions For Experienced

16. Write the code for "Hello world" using Express.

Create a new file by the name- index.js and type the following commands:

```
var express = require ('express');
var app = express ();

app.get ('/', function (req, res){
    res.send ("Hello world");
});

app.listen (3000);
```

Now go to the terminal after saving it and type:

17. What are the most used HTTP methods in Express JS?

The following HTTP methods are the most used ones:

- **GET** - A specified resource's representation is requested by the GET method. These requests can only retrieve data.
- **POST** - Posting of the data enclosed in the request as a new entity is done using the POST method. The entity is identified by the URI.
- **PUT** - Modification in the existing entity is done with the data enclosed in the request identified by the URI.
- **DELETE** - The request for deleting the specified source is made by the DELETE method.

18. How can the cookies be manipulated using 'Response.cookie()'?

We use the "res.cookie('username', 'Flavio')" command is used for manipulating. But it accepts a third parameter containing various options as specified below:

```
res.cookie ('username', 'Flavio', { domain: 'flaviocopes.com', path:
'/administrator', secure: true })
res.cookie ('username', 'Flavio' , { expires: new Date(Date.now() + 900000),
httpOnly: true})
```

19. When does a Cross-Origin resource get failed in Express JS?

A cross-Origin can fail in the following scenarios-

- If it's to a different domain
- If it's to a different port
- If it's to a different subdomain
- If it's to a different protocol.

20. How can you use a Pug template engine inside Express?

We will first install it using the given command:

```
npm install pug
```

Then we will set it as following when initializing the Express app:

```
const express = require ('express')
const app = express()
app.set ('view engine', 'pug')
```

21. What do you mean by the sanitizing input process?

People can always enter weird things via the client-side code. They use the tools to POST things directly to our endpoints. For this, the Express provides various sanitizing methods to prevent these happenings.

22. Mention some methods for sanitizing.

Consider the following sanitizing methods:

- Trim() will trim the characters at the beginning as well as the ending of a string.
- Escape() will replace ' , " , < , > , & , / with the corresponding HTML entities.
- NormalizeEmail() will canonicalize an email address.
- Blacklist() will remove the characters appearing on the blacklist.

23. Give an example of HTML form code allowing user to upload a file.

Consider the given example of HTML form code that allows a user to upload file.

```
<form method = "POST" action = "/submit-form">
  <input type = "file" name = "document" />
  <input type = "submit" />
</form>
```

24. What are the methods that you can call when the Formidable.File objects arise giving the information about the uploaded file?

We can call the following methods in such cases-

- **File.name**- the name of the file
- **File.path**- the path to which the file is written
- **File.size**- the size of the file in bytes
- **File. type**- the file's MIME-type.

25. What steps will you follow to set up HTTP for Express with the help of Let's Encrypt and Certbot?

We will follow the given steps to set up HTTP:

1. Installing certbot
2. Generating the SSL certificate by Certbot
3. Allowing Express to serve the static files
4. Confirming the domain
5. Obtaining the certificate
6. Setting up the renewal.

26. What options are available in the terminal command for generating a skeleton Express JS app?

The options available are given below:

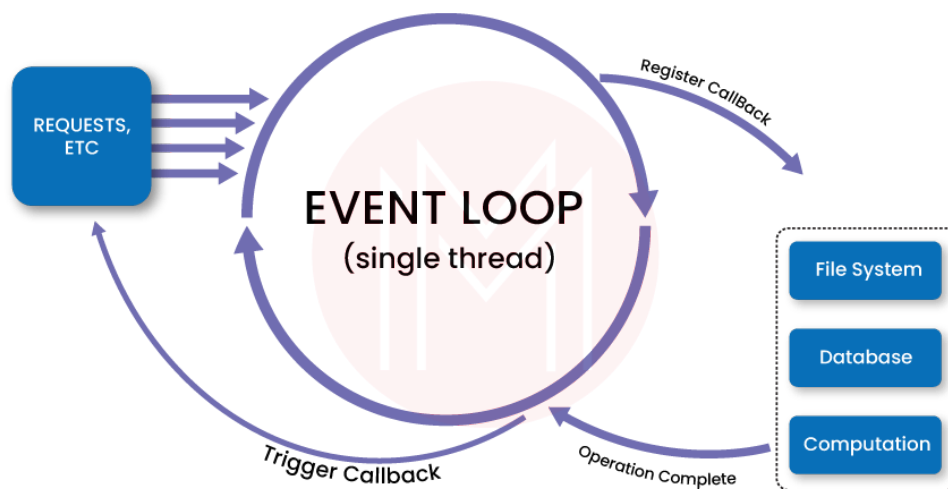
- --sessions or -s for adding session report
- --hogan or -H for adding Hogan.js engine support
- --ejs or -e for adding EJS engine support
- --css <engine> or -c <engine> for adding style sheet support

- -jshtml or -J for adding JSHTML engine support
- --force or -f for forcing app generation on the directory which is non-empty.

Related Article [Basic Examples of Node JS](#)

27. What is meant by an event-loop in Node JS?

The event-loop manages the async content using a listener and queue. The main thread sends the async function to a different thread whenever it requires to be executed. Alongside, v8 is allowed to execute the main code. The event loop has different stages including pending callbacks, timers, check, poll, close callbacks, etc. with different FIFO queues.



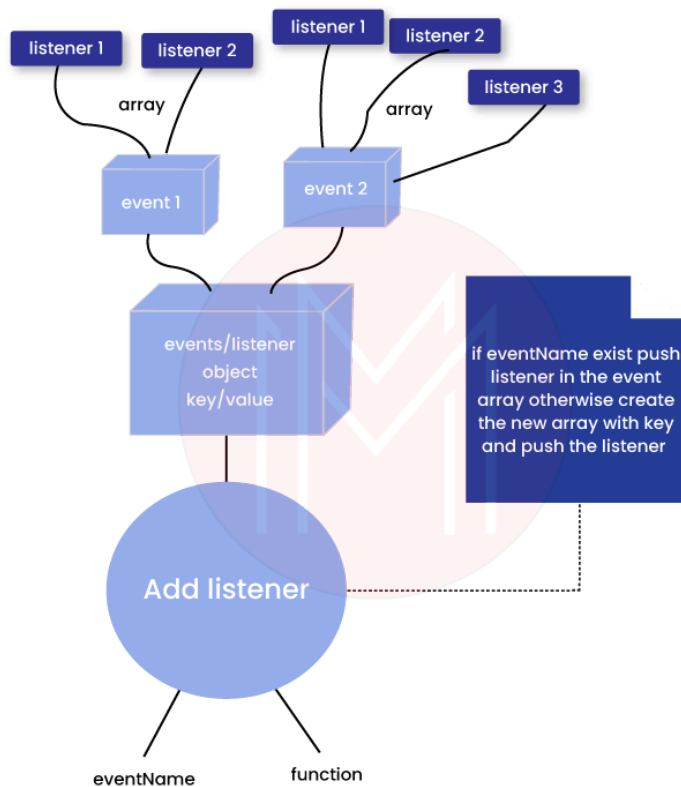
28. Why should the Express server and app be separated?

The server initializes the Middleware, routes, and other application logic. On the other hand, the app contains all the business logic that the server-initiated routes will serve. This enables the encapsulation of the business logic from the application logic for smooth functioning.

Check out [Node JS Interview Questions and Answers](#)

29. What is meant by an Event emitter in Node JS?

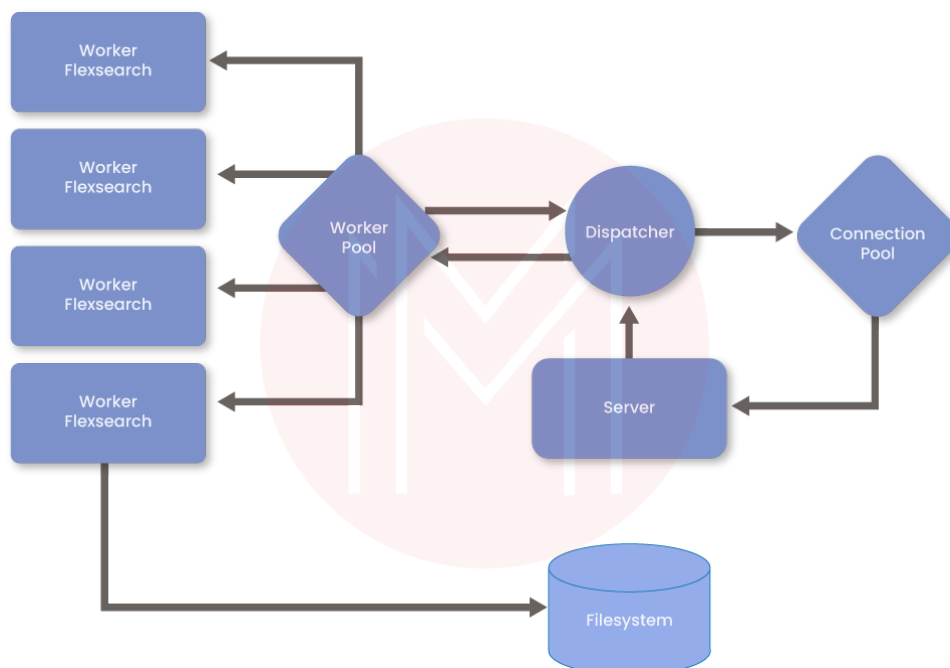
It's a class of Node JS capable of emitting events. We do this by attaching the named events emitted by the object by using the function- `eventEmitter.on()`.



30. Differentiate between worker threads and clusters in Node JS.

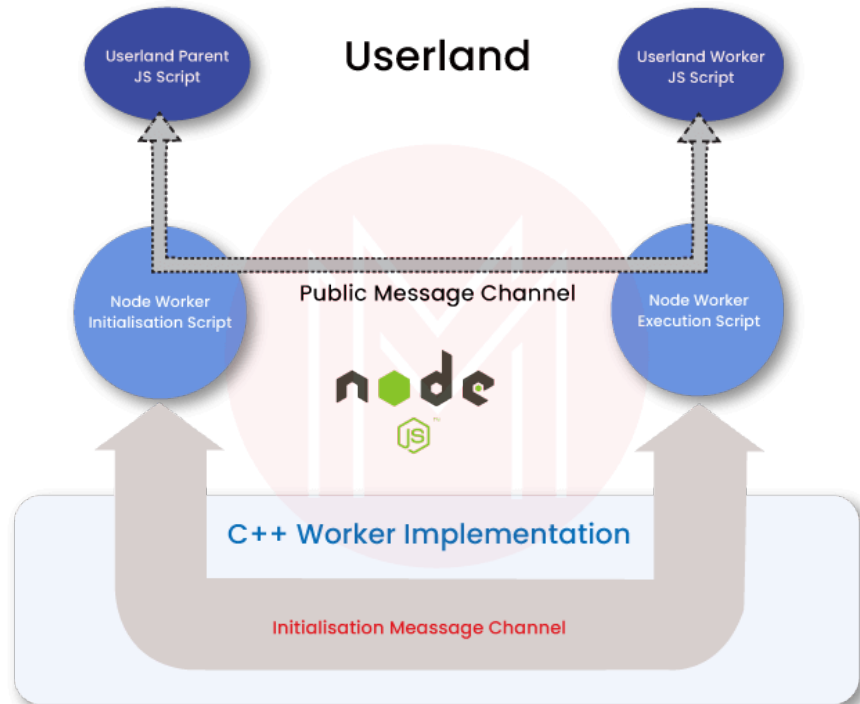
Cluster:

- Each CPU has one process with an IPC to communicate.
- Clusters help when multiple servers are required to accept HTTP requests through a single port.
- The processes have separate memory because of being spanned in different CPUs leading to memory issues.



Worker Threads:

- Only a single process is there with multiple threads.
- Each Node contains one Node having most APIs accessible.
- The memory is shared with other threads.
- We can use this for CPU-intensive tasks.



Let's go through the top 10 Frequently Asked Express JS Interview Questions and Answers (FAQs)

Most Common Express JS FAQs

1. List down major companies that use Express JS.

The following major companies use Express JS-

- Twitter
- Stack
- Accenture
- Intuit
- Bepro Company
- Trustpilot
- BlaBlaCar.

2. What are the popular alternatives to Express JS?

React JS, Meteor, Mean, Flask, Catalyst, Django, Apache Flex, and Laravel are some of the popular alternatives to Express JS.

3. Enlist some distinct features of Express JS.

Some distinct features of Express JS are given below-

- We can design single-page, multi-page, and hybrid web apps as well as APIs with Express JS.
- A routing table is defined for performing HTTP operations.
- Middleware can be set up for responding to RESTful/HTTP requests.
- The MVC-like structure enables organizing the web apps into MVC architecture.
- HTML pages can be dynamically rendered on the basis of passing arguments to templates.
- NoSQL, as well as RDBMS databases, are supported by it.
- High performance is delivered due to its super-fast I/O. the performance is adequate because of the thin layer prepared by it.
- Routing gets easy by its robust API.
- It is single-threaded as well as asynchronous.

4. Which major tools integrate with Express JS?

The following popular tools integrate with Express JS:

- Sentry
- Node JS
- Datadog
- Mean
- Nodemon
- Bugsnag
- LoopBack
- Sails JS.

5. Is Express JS a back-end framework or a front-end framework?

Express JS is a back-end framework built on JavaScript. It is the MEAN stack's back-end component. Here, 'M' refers to MongoDB, and it manages the database. 'E' refers to Express, and it manages the back-end. 'A' refers to AngularJS, and it handles the back-end. And 'N' refers to Node.

6. Differentiate between Node JS and Express JS.

Node JS is an open-source platform on which the JavaScript code is executed outside of a browser. It is used by several companies, including Uber, Walmart, Netflix, etc. It is a platform acting as a web server and not a programming language or framework. On the other hand, Express JS is a framework built on Node JS.

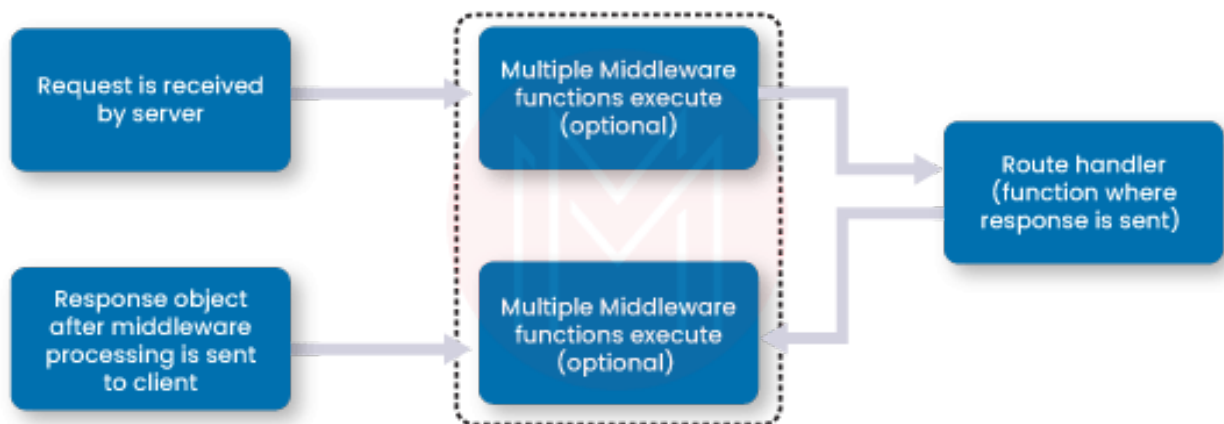
7. Which template engines are supported by Express?

All template engines following the locals, path, and callback signatures are supported by Express JS.

8. What is Middleware, and what are its functions?

Middleware is the function that we invoke before the final request process through the express routing layer. Its functions are given below:

1. Any code like setting headers, validation, etc., can be executed.
2. Changes can be made to the response (res) and request (req) objects.
3. The request-response cycle can also be ended by Middleware.
4. The next middleware function can be called in the stack for proceeding and processing the final request.



9. List the main types of Middleware.

The main types of Middleware are given below:

- Router-level Middleware
- Application-level Middleware
- Built-in Middleware
- Error-handling Middleware
- Third-party Middleware.

10. What is meant by Scaffolding in Express JS? What are the ways to achieve this?

The process of creating the structure of the application is referred to as scaffolding. The two ways of achieving it are given below:

- Express application generator
- Yeoman.

Conclusion

Node JS along with Express JS, allows building a complex and high-level web application from a single line of code easily. That's why it is one of the most sought-after skills in the industry today. And this was a comprehensive list of Express JS Interview Questions that are often asked in interviews. We hope that it will be beneficial for your scheduled interview in the near future.

—
—