

Express Js Interview Questions & Answers.

1)What is Express Js?

Express.js is a Node.js framework to develop the backend APIs of web and mobile application. It is Open Source NPM package and is generally installed after creating a Node.js application.

Node.js itself is a JavaScript framework and is mainly used to develop backend APIs. But it can be used in socket programming to create chat apps. In fact, Node.js can even be used without Express.js to create backend APIs. But the code is very complicated and more difficult to implement.

This is where Express.js shines and does the one job perfectly for which it was created, which is, creating API endpoints. It adds the functionality of middleware and routing with ease in a Node.js application.

2)How to install Express Js?

Since Express.js is a framework for Node.js, the basic requirement is to have Node.js installed on the system. The installation of Node.js can be done in many different ways, depending on the operating system. The best way is to go to the Node.js official site <https://Node.js.org/en/> and install it.

Node.js comes with NPM installed and it is required, because Express.js is an external node package, which needs to be added in a Node application.

3) Why to use Express Js?

Express has an in-built route support, so a developer can write responses to specific URLs with ease. We can also pass parameters to these routes, which add to the capabilities of the app. Express also supports multiple templating engines, through which you can generate static HTML for a web-browser. If you follow this approach, the pages will be generated by the server and there will be no need to write client code in Angular or React.

Without Express.js, we would have to write complex logic in Node.js, which is both, tedious and time consuming. Express.js code is minimalistic, efficient, and easily understandable by programmers, and because of it easily scalable.

4) 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:

- **req** - It is known as the request object, which is a callback function. In Express we receive the data from the client, using three different types of requests. They are req.params, req.query and req.body.
- **res** - It is also known as the response object. It is used to send the response back to the client. With the res object you can send many types of responses, like res.send, res.render, res.download and more.
- **next** - It is an optional callback function, which can be used to send the control to the next middleware or function.

5) What is scaffolding in Express Js ? What are the ways to achieve this?

Scaffolding is the process of creating a skeletal structure of a project. The production projects require a lot of modules to be added in the project before even starting to write the code. This is where scaffolding is used and there are many packages available for it in Express. We will now look at a scaffolding project. Express application generator & Yeoman.

6) 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.

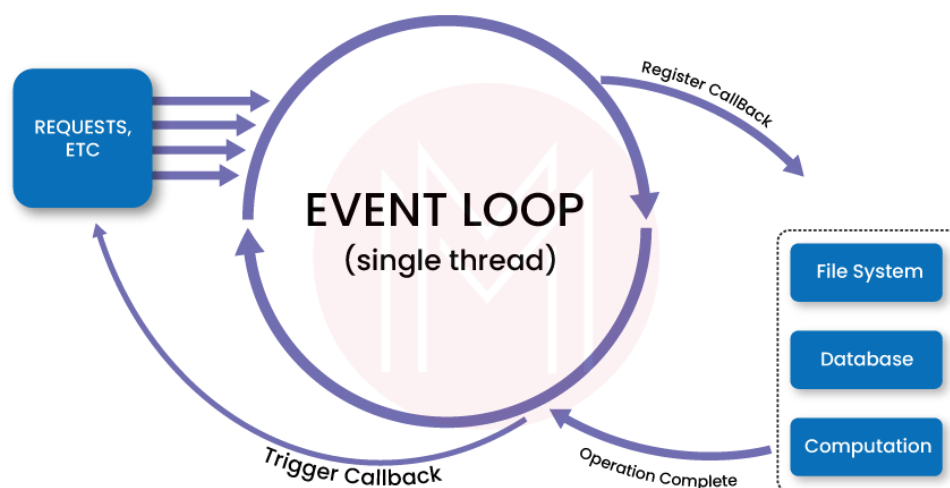
7) List the built-in middleware functions provided by Express.

Express JS provides the following built-in middleware functions:

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

8) 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.



9) 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.

10) 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.

11) 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.

12) 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.

13) 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.

14) 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.

15) 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.

16) 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:

- i. Express cors module
- ii. Res.header() (or res.set()): Multiple headers can be set using this way.
- iii. Res.setHeader(): Only a single header can be set in this way.

17) Explain the detail use of response object in Express ?

The response object or the res object, is used to send the data back to the client. It generally happens after the request is sent from the client in the req object. So, it is a part of the request and response cycle.

Now, with the res object we have many methods like send(), status(), append(), redirect() and render().

18) 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.

19) 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.

20) 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.

Follow @Onkar Deolankar for more such content !!!!!!!