

Important Express.js Interview Questions and Answers

1. What is Express.js?

Express.js is a fast, unopinionated, and minimalist web framework for Node.js used to build web applications and APIs.

2. What are the main features of Express.js?

Features include middleware support, routing, template engines, HTTP utility methods, and error handling.

3. How do you install Express.js?

You can install it using npm: `npm install express --save`.

4. What is middleware in Express.js?

Middleware functions are functions that have access to the request and response objects, and can modify them or end the request-response cycle.

5. What are types of middleware in Express.js?

Types include application-level middleware, router-level middleware, built-in middleware, and error-handling middleware.

6. How do you define routes in Express.js?

You can define routes using methods like `app.get()`, `app.post()`, `app.put()`, and `app.delete()`.

7. What is the difference between app.use() and app.get()?

`app.use()` is used to mount middleware, while `app.get()` defines a route handler for GET requests.

8. How can you serve static files in Express.js?

Use the built-in middleware: `app.use(express.static('public'))`.

9. What is the purpose of the next() function in Express.js?

The `next()` function passes control to the next middleware function in the stack.

10. How can you handle errors in Express.js?

Define an error-handling middleware function with four parameters: `(err, req, res, next)`.

11. How do you set up a basic Express.js server?

Example: `const express = require('express'); const app = express(); app.listen(3000);`

12. How do you handle JSON data in Express.js?

Use `app.use(express.json())` to parse incoming JSON requests.

13. What is express.Router()?

`express.Router()` is used to create modular route handlers and manage routes in separate files.

14. How do you implement CORS in Express.js?

Install and use the cors package: `npm install cors` then `app.use(require('cors')())`.

15. How do you connect a database (like MongoDB) to Express.js?

You can use Mongoose or MongoDB driver, e.g.,
`mongoose.connect('mongodb://localhost/dbname')`.

16. What is the purpose of app.listen()?

`app.listen()` starts the Express server and listens for incoming requests on a specified port.

17. How can you use environment variables in Express.js?

Use the dotenv package to load environment variables from a `.env` file: `require('dotenv').config()`.

18. What is the difference between res.send() and res.json()?

`res.send()` can send various types of responses, while `res.json()` sends a JSON-formatted response.

19. How can you redirect a route in Express.js?

Use `res.redirect('/path')` to redirect the client to a different route.

20. How can you secure an Express.js app?

By using Helmet, CORS, input validation, HTTPS, and secure cookie handling.