

## Nodejs Questions and Answers

**1. What does req.body contain?**

- A. Route parameters
- B. Response data
- C. Parsed request payload (e.g. JSON)
- D. URL query parameters

c

**2. Which of the following can be used to hash passwords in Node.js?**

- A. jsonwebtoken
- B. bcrypt
- C. crypto-js
- D. mongoose

b

**3. What is the purpose of express.Router()?**

- A. Launch the server
- B. Create sub-routers/modular routes
- C. Connect to the database
- D. Enable CORS

b

**4. Which status code represents unauthorized access?**

- A. 403
- B. 404
- C. 401
- D. 500

c

**5. What method is used to terminate the response cycle in Express?**

- A. res.write()
- B. res.end()
- C. res.send()
- D. res.destroy()

b

**6. What is the correct order of middleware execution in Express?**

- A. Last defined to first defined
- B. Only custom middleware runs
- C. Top-down as defined in the code
- D. Random order

c

**7. What does res.json() do?**

- A. Sets header only
- B. Sends a raw file
- C. Ends the response with JSON data
- D. Parses JSON request

c

**8. Which header is commonly used for bearer token authorization?**

- A. x-auth
- B. Authorization
- C. Auth
- D. X-Token

b

**9. What's the use of app.all('\*', handler) in Express?**

- A. Handle static assets
- B. Catch-all for all methods and routes
- C. Disable all routes
- D. Add headers

b

**10. In Express, how can you define an error-handling middleware?**

- A. (req, res)
- B. (err, req, res, next)
- C. (res, req)
- D. (next, err)

B

**11. Which command installs all project dependencies defined in package.json?**

- A. npm update
- B. npm install
- C. node install
- D. npm create

b

**12. What is the use of nodemon in development?**

- A. Build scripts
- B. Code formatting
- C. Automatically restarts server on file changes
- D. Debugging errors

c

**13. What is middleware chaining in Express?**

- A. Defining multiple databases
- B. Serving multiple HTML files
- C. Executing multiple middleware functions in sequence
- D. Creating chained routes

c

**14. What is the best place to store environment variables in a Node.js project?**

- A. .env file
- B. config.js
- C. server.js
- D. README.md

a

**15. What command is used to run unit tests with Jest?**

- A. jest.run()
- B. npm start
- C. npm test
- D. npm jest

c

**16. Which command is used to initialize a new Node.js project?**

- A. node init
- B. npm install
- C. npm init
- D. node start

c

**17. Which of the following is a valid way to define middleware only for one route?**

- A. app.use() globally
- B. Use inside app.get() or app.post()
- C. app.middleware()
- D. app.route() only

b

**18. What status code typically represents "Forbidden"?**

- A. 400
- B. 403
- C. 401
- D. 500

b

**19. Which Node.js core module provides utilities for debugging and formatting?**

- A. fs
- B. util
- C. events
- D. stream

b

**20. What does process.env.PORT typically represent?**

- A. The client's port number
- B. The database port
- C. The environment-defined server port
- D. A hardcoded port

c

**21. What method is used in Mongoose to delete a document by ID?**

- A. delete()
- B. remove()
- C. findByIdAndDelete()
- D. deleteById()

c

**22. In Express, how do you return a 500 Internal Server Error manually?**

- A. res.error(500)
- B. res.sendStatus(500)
- C. res.throw(500)
- D. res.status(404)

b

**23. What does app.listen(PORT, callback) do in Express?**

- A. Binds route handlers
- B. Starts listening for HTTP requests
- C. Logs to console
- D. Starts a database connection

b

**24. How do you make Express serve static files?**

- A. app.static()
- B. express.static()
- C. res.static()
- D. app.useStatic()

b

**25. What will res.status(204).send() do?**

- A. Return OK response
- B. Return a JSON object
- C. Return a no-content response
- D. Return an error

c

**26. What is the main reason for using environment variables?**

- A. Faster performance
- B. Separate config from code
- C. Better logging
- D. Avoid JSON parsing

b

**27. What is the recommended way to structure an Express app for scalability?**

- A. Single file
- B. Monolithic structure
- C. MVC (Model-View-Controller) pattern
- D. Flat file system

c

**28. Which method is used to apply middleware only to a router in Express?**

- A. router.apply()
- B. router.use()
- C. router.route()
- D. router.listen()

b

**29. What is res.render() used for?**

- A. Send plain text
- B. Render JSON
- C. Render templates with a view engine
- D. Convert HTML to JSON

c

**30. What does req.headers contain?**

- A. The server response
- B. The incoming HTTP request headers
- C. The rendered view
- D. The route definition

B

**31. How can you connect to MongoDB using Mongoose?**

- A. mongoose.link()
- B. mongoose.start()
- C. mongoose.connect()
- D. mongoose.open()

c

**32. What is Node.js primarily used for?**

- A. Designing UI components
- B. Running server-side applications
- C. Managing relational databases
- D. Compiling Java programs

b

**33. Which module in Node.js is used to work with the file system?**

- A. http
- B. os
- C. fs
- D. net

c

**34. What does the npm init command do?**

- A. Installs a new package
- B. Creates a package.json file
- C. Starts the Node server
- D. Initializes a Git repository

b

**35. What is the default port number for a Node.js app if not specified?**

- A. 8080
- B. 3000
- C. 80
- D. No default

D

**36. What is Express.js?**

- A. A database
- B. A Node.js framework
- C. A programming language
- D. A compiler

b

**37. Which method is used to define a GET route in Express?**

- A. app.route()
- B. app.get()
- C. app.listen()
- D. app.fetch()

b

**38. What middleware is used to parse JSON in Express?**

- A. body-parser
- B. express.json()
- C. express.parser()
- D..urlencoded()

b

**39. Which HTTP status code represents “Resource Not Found”?**

- A. 200
- B. 201
- C. 404
- D. 500

c

**40. How do you start a Node.js server file named app.js?**

- A. start app.js
- B. node app.js
- C. npm run app.js
- D. node start app.js

B

**41. What is a middleware function in Express?**

- A. A route handler
- B. A static file
- C. A function that handles requests and responses before the final handler
- D. A database connector

C

**42. Which function is used to listen for incoming requests in Express?**

- A. app.handle()
- B. app.start()
- C. app.listen()
- D. app.run()

C

**43. What is the purpose of req.params in Express?**

- A. Access request body
- B. Access query strings
- C. Access URL route parameters
- D. Access cookies

C

**44. What does the res.send() method do in Express?**

- A. Ends the request
- B. Sends an HTTP request
- C. Sends a response to the client
- D. Logs a message

C

**45. What does res.status(404).json({...}) do?**

- A. Sends HTML
- B. Redirects to another route
- C. Sends a 404 error response in JSON
- D. Parses incoming requests

C

**46. Which tool is commonly used to test REST APIs?**

- A. MongoDB
- B. Postman
- C. Visual Studio
- D. Redis

b

**47. In REST APIs, which method is typically used to delete a resource?**

- A. GET
- B. POST
- C. DELETE
- D. PUT

c

**48. What is the primary role of the cors package in Express?**

- A. Protect from XSS
- B. Enable cross-origin requests
- C. Validate request body
- D. Parse JSON

b

**49. What status code indicates a successful creation of a resource?**

- A. 200
- B. 201
- C. 204
- D. 301

b

**50. Which Mongoose method is used to retrieve all documents in a collection?**

- A. findAll()
- B. getAll()
- C. get()
- D. find()

D

**51. Which method is used to update a document in MongoDB using Mongoose?**

- A. Model.find()
- B. Model.save()
- C. Model.updateOne()
- D. Model.create()

C

**52. What does app.route('/users') allow you to do?**

- A. Chain multiple route handlers
- B. Use query parameters
- C. Define global middleware
- D. Serve static assets

a

**53. How do you define a dynamic route in Express?**

- A. /user?id=123
- B. /user::id
- C. /user/:id
- D. /user{}

C

**54. Which module is used to manage file uploads in Express?**

- A. body-parser
- B. fs
- C. multer
- D. sharp

C

**55. Which object holds query string parameters in Express?**

- A. req.params
- B. req.body
- C. req.query
- D. req.route

C

**56. What does `app.use()` do in Express?**

- A. Serves static files
- B. Registers a middleware
- C. Initializes the server
- D. Handles GET requests

b

**57. Which of the following can be used to structure MongoDB models in Node.js?**

- A. Sequelize
- B. Mongoose
- C. TypeORM
- D. Prisma

b

**58. What does `dotenv` module help with?**

- A. Creating HTML templates
- B. Loading environment variables
- C. Encrypting passwords
- D. Serving static files

b

**59. What does the `next()` function do in Express middleware?**

- A. Ends the server
- B. Moves to the next middleware or route
- C. Stops the request
- D. Redirects to home page

b

**60. What is the main purpose of Mongoose in a Node.js application?**

- A. Serve static HTML
- B. Connect to PostgreSQL
- C. Define MongoDB schemas and models
- D. Encrypt passwords

C

**61. What are common status codes you return in a REST API?**

Code	Meaning
200	OK
201	Created
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

**62. What is the difference between synchronous and asynchronous code in Node.js?**

Synchronous code blocks the event loop until execution is complete.

Asynchronous code uses callbacks, promises, or async/await to perform non-blocking operations.

**63. How do you organize a scalable Express.js project structure?**

A typical folder structure:

```
/controllers  
/routes  
/models  
/middleware  
/utils  
/app.js  
/server.js
```

**64. What is CORS and how do you enable it in Express?**

CORS (Cross-Origin Resource Sharing) is a security feature that restricts cross-origin HTTP requests.

Enable with the cors middleware:

```
const cors = require('cors');  
app.use(cors());
```

**65. What is the difference between PUT and PATCH?**

PUT: Full update, replaces the entire resource.

PATCH: Partial update, modifies specific fields.

## 66. How do you handle asynchronous code in Express routes?

Use async/await and catch errors with a wrapper or try/catch:

```
app.get('/users', async (req, res, next) => {
  try {
    const users = await User.find();
    res.json(users);
  } catch (err) {
    next(err);
  }
});
```

## 67. What is the purpose of express.Router()?

express.Router() is used to create modular, mountable route handlers. It helps split the application into multiple route files and organize the code better.

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

res.send() can send any type of content.

res.json() is specifically used to send JSON-formatted responses and automatically sets the content-type to application/json.

## 69. What is the use of next() in Express middleware?

next() passes control to the next middleware function in the stack. Without calling next(), the request will be left hanging and the response may never be sent.

## 70. How do you create a basic server in Express?

```
const express = require('express');
const app = express();
app.get('/', (req, res) => res.send('Hello World!'));
app.listen(3000, () => console.log('Server running on port 3000'));
```

## 71. What are middleware functions in Express.js?

Middleware functions in Express are functions that have access to the request (req), response (res), and the next() function. They are used for tasks like logging, authentication, parsing, error handling, etc.

```
app.use((req, res, next) => {
  console.log('Middleware triggered');
  next();
});
```

**72. What is the difference between require() and import?**

require() is the CommonJS module system, used in traditional Node.js environments.

import is the ES6 module syntax, used in modern JavaScript (and supported in Node.js via .mjs files or "type": "module" in package.json).

**73. What is Express.js?**

Express.js is a minimal and flexible Node.js web application framework that provides a robust set of features to build web and mobile applications, such as routing, middleware, templating, and error handling.

**74. What is Node.js?**

Node.js is an open-source, cross-platform JavaScript runtime environment that allows you to run JavaScript code outside the browser. It is built on Chrome's V8 engine and is designed for building scalable network applications using non-blocking I/O and an event-driven architecture.