1. Introduction to Client-Server Architecture

1. **Which of the following best describes client-server architecture?** A) Both client and server are on the same machine
   B) Clients request services, and servers provide them
   C) Server only serves the requests of other servers
   D) Clients provide services, and servers request them

   Answer: B) Clients request services, and servers provide them

2. **In a client-server model, the server is responsible for which of the following?** A) Receiving requests and sending responses
   B) Initiating requests to clients
   C) Generating client requests
   D) Displaying the UI

   Answer: A) Receiving requests and sending responses

3. **Which layer of the OSI model does the server typically operate on?** A) Application layer
   B) Transport layer
   C) Network layer
   D) Data Link layer

   Answer: A) Application layer

4. **What protocol is commonly used for communication between client and server on the web?** A) FTP
   B) HTTP
   C) SMTP
   D) Telnet

   Answer: B) HTTP

5. **Which of the following can be used as a client in a client-server architecture?** A) Web browser
   B) File server
   C) Database
   D) DNS Server

   Answer: A) Web browser

---

2. Understanding Server and Request Handling

6. **What does the server do when it receives a request from a client?** A) Processes the request and returns a response
   B) Ignores the request

C) Sends a new request to another server
D) Analyzes the request and forwards it to the database

Answer: A) Processes the request and returns a response

7. **Which HTTP method is used by the server to send data back to the client?** A) POST
B) PUT
C) GET
D) Response

Answer: C) GET

8. **In Node.js, which object is used to handle HTTP requests and send responses?**
A) `response`
B) `request`
C) `res`
D) `req`

Answer: B) `request`

---

3. Understanding Node.js

9. **What is Node.js built on?** A) JavaScript
B) Python
C) Java
D) Ruby

Answer: A) JavaScript

10. **Which of the following is NOT an advantage of Node.js?** A) High performance for I/O operations
B) Uses event-driven, non-blocking I/O
C) Great for CPU-intensive tasks
D) Handles concurrent requests effectively

Answer: C) Great for CPU-intensive tasks

11. **Node.js is particularly well-suited for applications that require:** A) Heavy computation
B) High-frequency trading
C) Real-time data processing
D) Complex graphical rendering

Answer: C) Real-time data processing

12. **What type of architecture does Node.js use for handling multiple requests?** A) Multi-threaded

B) Single-threaded with event loop
C) Multi-process
D) Pre-emptive scheduling

Answer: B) Single-threaded with event loop

4. Installing Node.js and Creating a Node.js Server

13. **Which of the following commands is used to install Node.js on a system?** A) `npm install node`
    B) `sudo apt install nodejs`
    C) `install nodejs`
    D) `npm install`

    Answer: B) `sudo apt install nodejs`

14. **To create a simple Node.js HTTP server, which built-in module is required?** A) `express`
    B) `http`
    C) `mongoose`
    D) `fs`

    Answer: B) `http`

15. **What is the purpose of the `server.listen()` method in Node.js?** A) To send a response to the client
    B) To close the server
    C) To make the server start accepting requests
    D) To handle incoming data

    Answer: C) To make the server start accepting requests

5. Modules and npm

16. **What is the role of `npm` in Node.js?** A) To create new modules
    B) To handle the installation of third-party packages
    C) To run Node.js applications
    D) To debug Node.js code

    Answer: B) To handle the installation of third-party packages

17. **Which command is used to initialize a new Node.js project and create a `package.json` file?** A) `npm install`
    B) `npm init`
    C) `npm start`
    D) `npm update`

Answer: B) `npm init`

18. **Which file contains the dependencies for a Node.js project?** A) `package.json`
    B) `node_modules.json`
    C) `server.js`
    D) `index.html`

    Answer: A) `package.json`

---

6. Handling Static Pages and File Streams

19. **In Node.js, how do you serve static files like images or CSS?** A) Using the `fs` module
    B) Using the `express.static()` middleware
    C) Using the `http` module
    D) Using `fs.readFile()`

    Answer: B) Using the `express.static()` middleware

20. **Which module can be used to read or write files in Node.js?** A) `http`
    B) `path`
    C) `fs`
    D) `express`

    Answer: C) `fs`

---

7. Introduction to Express and Routing

21. **What is Express.js used for in Node.js?** A) To manage databases
    B) To handle HTTP requests and simplify routing
    C) To create static pages
    D) To monitor server performance

    Answer: B) To handle HTTP requests and simplify routing

22. **What is the default port number for an Express app?** A) 3000
    B) 5000
    C) 8080
    D) 8000

    Answer: A) 3000

23. **Which of the following methods in Express is used to handle GET requests?** A) `app.post()`
    B) `app.get()`

C) `app.delete()`
D) `app.put()`

Answer: B) `app.get()`

24. **Which of the following is the correct syntax for defining a route parameter in Express?** A) `app.get("/users/:id", ...)`
B) `app.get("/users/{id}", ...)`
C) `app.route("/users/:id", ...)`
D) `app.path("/users/id", ...)`

Answer: A) `app.get("/users/:id", ...)`

---

8. Middleware in Express

25. **What is the purpose of middleware in Express?** A) To handle data storage
B) To log incoming requests
C) To handle routing
D) To manage databases

Answer: B) To log incoming requests

26. **Which of the following is an example of application-level middleware in Express?** A) `app.use(express.json())`
B) `app.get("/home", ...)`
C) `express.static("public")`
D) `app.listen(3000)`

Answer: A) `app.use(express.json())`

---

9. Authentication, Authorization, and Sessions

27. **What is the purpose of JSON Web Tokens (JWT) in authentication?** A) To encrypt data in databases
B) To transfer session data securely between client and server
C) To handle database queries
D) To store user credentials

Answer: B) To transfer session data securely between client and server

28. **Which of the following is used to manage user sessions in Express?** A) `session` module
B) `passport` module
C) `jsonwebtoken` module
D) `cookie-parser`

Answer: A) `session` module

29. **What is the difference between authentication and authorization?** A) Authentication verifies identity, authorization determines access level
    B) Authentication determines access level, authorization verifies identity
    C) Authentication and authorization are the same
    D) Authentication is for administrators, authorization is for users

    Answer: A) Authentication verifies identity, authorization determines access level

---

10. EJS Template Engine and Database Integration

30. **What is EJS used for in Node.js applications?** A) To handle HTTP requests
    B) To connect to databases
    C) To generate dynamic HTML views
    D) To manage sessions

    Answer: C) To generate dynamic HTML views

31. **Which of the following is used to connect a Node.js application to MongoDB?** A) Sequelize
    B) Mongoose
    C) Express.js
    D) Knex.js

    Answer: B) Mongoose

---

11. Introduction to Databases (SQL and NoSQL)

32. **Which of the following is a characteristic of NoSQL databases?** A) Data is stored in tables with rows and columns
    B) Data can be unstructured or semi-structured
    C) NoSQL databases use SQL queries
    D) Data is relational

    Answer: B) Data can be unstructured or semi-structured

33. **Which of the following is an example of a relational database?** A) MongoDB
    B) MySQL
    C) Cassandra
    D) Redis

    Answer: B) MySQL

---

12. Blocking vs Non-blocking Code in Node.js

34. **What is the primary difference between blocking and non-blocking code in Node.js?** A) Blocking code uses multiple threads, non-blocking code uses a single thread
    B) Blocking code cannot be executed, while non-blocking code can
    C) Blocking code halts program execution, while non-blocking code allows other tasks to run concurrently
    D) Blocking code processes requests asynchronously, while non-blocking code processes them synchronously

Answer: C) Blocking code halts program execution, while non-blocking code allows other tasks to run concurrently

35. **Which of the following operations in Node.js is typically non-blocking?** A) Reading files synchronously
    B) Writing files synchronously
    C) Handling HTTP requests asynchronously
    D) Executing SQL queries synchronously

Answer: C) Handling HTTP requests asynchronously

---

13. Introduction to Multer and File Handling

36. **What is the purpose of the Multer library in Node.js?** A) To parse JSON data in HTTP requests
    B) To handle file uploads in Express applications
    C) To interact with MongoDB
    D) To parse cookies

Answer: B) To handle file uploads in Express applications

37. **Which of the following is a correct way to configure Multer for disk storage?** A) `multer.diskStorage()`
    B) `multer.memoryStorage()`
    C) `multer.createStorage()`
    D) `multer.config()`

Answer: A) `multer.diskStorage()`

38. **Which of the following properties is used to specify the file destination in Multer's disk storage configuration?** A) `destination`
    B) `output`
    C) `upload`
    D) `filePath`

Answer: A) `destination`

39. **When handling file uploads with Multer, which event is emitted when a file is successfully uploaded?** A) `error`
    B) `file`
    C) `upload`
    D) `finish`

Answer: B) `file`

---

14. Session Management in Node.js

40. **What module is commonly used for session management in Express?** A) `express-session`
    B) `passport`
    C) `jsonwebtoken`
    D) `cookie-parser`

Answer: A) `express-session`

41. **Which of the following methods is used to store session data in a memory store?**
    A) `session.memory()`
    B) `session.set()`
    C) `session.store()`
    D) `session.use()`

Answer: B) `session.set()`

42. **Which of the following is true about sessions in Express.js?** A) They are automatically created when a user visits the website for the first time
    B) Sessions are stored by default in a database
    C) Sessions allow data to persist between requests from the same client
    D) Sessions are created manually by the developer

Answer: C) Sessions allow data to persist between requests from the same client

---

15. Authentication and Authorization

43. **Which of the following methods is used for user authentication in Express?** A) Middleware
    B) `passport.authenticate()`
    C) `express-session()`
    D) `jsonwebtoken.verify()`

Answer: B) `passport.authenticate()`

44. **What is the role of an authentication token, such as a JSON Web Token (JWT)?**
    A) To store user passwords securely
    B) To verify the identity of a user
    C) To store session data
    D) To manage database queries

Answer: B) To verify the identity of a user

45. **What is the purpose of `passport` in Node.js authentication?** A) To handle authorization logic
    B) To manage sessions
    C) To authenticate users by using different strategies (local, OAuth, etc.)
    D) To create secure tokens for authentication

Answer: C) To authenticate users by using different strategies (local, OAuth, etc.)

46. **Which of the following is a common strategy for user authentication in Node.js applications?** A) Basic Authentication
    B) OAuth
    C) JWT (JSON Web Token)
    D) All of the above

Answer: D) All of the above

---

16. JSON Web Tokens (JWT) and Authentication

47. **Which of the following is a part of a JSON Web Token?** A) Payload
    B) Signature
    C) Header
    D) All of the above

Answer: D) All of the above

48. **What does the "exp" claim in a JSON Web Token signify?** A) The time the token was created
    B) The expiration time of the token
    C) The issuer of the token
    D) The user associated with the token

Answer: B) The expiration time of the token

49. **Which of the following statements is true regarding JWT tokens?** A) They are encrypted and stored in a database
    B) They are stateless and do not require server-side storage
    C) They can only be used once
    D) They are stored in cookies by default

Answer: B) They are stateless and do not require server-side storage

50. **In which part of the HTTP request is a JWT typically sent?** A) Request body
    B) Request header (Authorization)
    C) Query string
    D) Cookie

Answer: B) Request header (Authorization)

---

17. Client-Side Rendering (CSR) vs Server-Side Rendering (SSR)

51. **Which of the following is a key benefit of Server-Side Rendering (SSR)?** A) Faster page load times due to rendering on the client side
    B) Reduced load on the server
    C) Improved SEO performance by serving pre-rendered HTML to search engines
    D) Easier to implement on static websites

Answer: C) Improved SEO performance by serving pre-rendered HTML to search engines

52. **Which of the following is true about Client-Side Rendering (CSR)?** A) It provides better SEO out-of-the-box
    B) It requires an initial server request to fetch HTML, CSS, and JavaScript
    C) It is typically used in static websites
    D) It renders HTML on the server before sending it to the client

Answer: B) It requires an initial server request to fetch HTML, CSS, and JavaScript

53. **Which technology is commonly associated with Server-Side Rendering in React?**
    A) React Router
    B) Next.js
    C) Express.js
    D) Webpack

Answer: B) Next.js

---

18. Template Engines in Node.js (EJS)

54. **What is the role of a template engine in Node.js applications?** A) To compile server-side code into HTML
    B) To manage database queries
    C) To route HTTP requests
    D) To serve static files

Answer: A) To compile server-side code into HTML

55. **Which of the following is true about EJS templates in Node.js?** A) EJS does not support embedding JavaScript inside HTML
    B) EJS allows you to include partial views like headers or footers in templates

C) EJS templates cannot receive dynamic data
D) EJS is only used for client-side rendering

Answer: B) EJS allows you to include partial views like headers or footers in templates

56. **How do you pass data from an Express route to an EJS template?** A) Using `res.send()`
    B) Using `res.render()` with a data object
    C) Using `res.json()`
    D) Using `app.use()`

Answer: B) Using `res.render()` with a data object

57. **What is the purpose of partials in EJS?** A) To render entire HTML pages
    B) To create reusable components (e.g., header, footer)
    C) To store dynamic data
    D) To define new routes

Answer: B) To create reusable components (e.g., header, footer)

---

19. Introduction to Databases: SQL vs NoSQL

58. **Which of the following is an example of a NoSQL database?** A) MySQL
    B) PostgreSQL
    C) MongoDB
    D) SQLite

Answer: C) MongoDB

59. **Which of the following is a characteristic of SQL databases?** A) Data is stored in a non-tabular format
    B) Data is organized into collections
    C) Data can be queried using SQL queries
    D) SQL databases are schema-less

Answer: C) Data can be queried using SQL queries

60. **Which of the following is NOT a feature of NoSQL databases?** A) Schema-less design
    B) Horizontal scalability
    C) Support for complex joins
    D) Flexible data models

Answer: C) Support for complex joins

---

20. Connecting MongoDB with Node.js

61. **Which of the following packages is used to connect a Node.js application with MongoDB?** A) mongoose
    B) mongodb-driver
    C) Sequelize
    D) knex

Answer: A) mongoose

62. **Which of the following is required to create a connection with MongoDB using Mongoose?** A) `mongoose.connect(<url>, <options>)`
    B) `mongoose.db(<url>, <options>)`
    C) `mongoose.connect()`
    D) `mongoose.connectToDB(<url>)`

Answer: A) `mongoose.connect(<url>, <options>)`

63. **What is a MongoDB document equivalent to in SQL?** A) Table
    B) Row
    C) Column
    D) Query

Answer: B) Row

21. Handling Errors in Node.js

64. **Which of the following is the correct way to handle exceptions in Node.js?** A) Using `try-catch` blocks
    B) Using `catch` function
    C) Using `throw` without a try block
    D) Node.js does not support error handling

Answer: A) Using `try-catch` blocks

65. **What does the `uncaughtException` event in Node.js handle?** A) Errors during the event loop
    B) Errors that occur asynchronously
    C) Errors that are not caught by try-catch blocks
    D) Syntax errors in the code

Answer: C) Errors that are not caught by try-catch blocks

66. **Which of the following is used to handle errors in Express middleware?** A) `next()`
    B) `res.send()`
    C) `throw`
    D) `app.error()`

Answer: A) `next()`

67. **What is the purpose of the `next()` function in Express?** A) To terminate the request
    B) To invoke the next middleware function in the stack
    C) To throw an error
    D) To send a response to the client

Answer: B) To invoke the next middleware function in the stack

---

22. Working with Asynchronous Code in Node.js

68. **Which of the following methods is used to handle asynchronous operations in Node.js?** A) Callbacks
    B) Promises
    C) `async/await`
    D) All of the above

Answer: D) All of the above

69. **Which of the following is a disadvantage of using callbacks for handling asynchronous code?** A) Callbacks are slower than promises
    B) Callbacks can lead to "callback hell" (nested callbacks)
    C) Callbacks do not work with asynchronous functions
    D) Callbacks are not supported in Node.js

Answer: B) Callbacks can lead to "callback hell" (nested callbacks)

70. **Which of the following is true about Promises in JavaScript?** A) They are used to handle synchronous operations
    B) They allow chaining of multiple asynchronous operations
    C) They make asynchronous code execute sequentially
    D) They do not return any value

Answer: B) They allow chaining of multiple asynchronous operations

71. **What does the `async` keyword in front of a function indicate?** A) The function is executed synchronously
    B) The function will always return a promise
    C) The function will return a generator function
    D) The function will run in the background

Answer: B) The function will always return a promise

72. **What does the `await` keyword do in JavaScript?** A) Pauses the execution of the function until a Promise is resolved
    B) Executes a function asynchronously
    C) Makes a function synchronous
    D) Returns a resolved promise

Answer: A) Pauses the execution of the function until a Promise is resolved

---

23. Express Routing Methods and Parameters

73. **Which HTTP method is used to retrieve information from a server in Express?**
    A) POST
    B) PUT
    C) GET
    D) DELETE

Answer: C) GET

74. **What is the purpose of route parameters in Express?** A) To define different paths for each route
    B) To accept dynamic values in routes
    C) To handle HTTP request methods
    D) To generate the response from the server

Answer: B) To accept dynamic values in routes

75. **Which of the following is a correct example of a route with a parameter in Express?** A) `app.get("/users/:id", (req, res) => {...})`
    B) `app.get("/users/{id}", (req, res) => {...})`
    C) `app.get("/users[id]", (req, res) => {...})`
    D) `app.get("/users?id", (req, res) => {...})`

Answer: A) `app.get("/users/:id", (req, res) => {...})`

76. **How can you capture query parameters in Express?** A) `req.body`
    B) `req.query`
    C) `req.params`
    D) `req.headers`

Answer: B) `req.query`

---

24. Advanced Express Middleware

77. **What is the function of router-level middleware in Express?** A) It is used to handle requests for specific routes
    B) It is used to log server requests globally
    C) It processes requests before they reach the route handler
    D) It handles errors across all routes

Answer: A) It is used to handle requests for specific routes

78. **What is an example of error-handling middleware in Express?** A) `app.use(errorHandler)`
    B) `app.get("/path", errorHandler)`
    C) `app.use((err, req, res, next) => {...})`
    D) `app.get((err, req, res) => {...})`

Answer: C) `app.use((err, req, res, next) => {...})`

79. **Which of the following is true about third-party middleware in Express?** A) It is always used for logging HTTP requests
    B) It is pre-built middleware that is created by other developers
    C) It cannot be customized
    D) It only runs after application-level middleware

Answer: B) It is pre-built middleware that is created by other developers

---

25. Working with Static Files in Express

80. **Which middleware function is used to serve static files in Express?** A) `express.static()`
    B) `app.static()`
    C) `serveStaticFiles()`
    D) `app.use(static())`

Answer: A) `express.static()`

81. **How do you set the public directory for serving static files in Express?** A) `app.use(express.static('public'))`
    B) `app.static('public')`
    C) `express.static('/public')`
    D) `app.serve('public')`

Answer: A) `app.use(express.static('public'))`

82. **Which of the following files can be served as static files in Express?** A) HTML, CSS, JavaScript files
    B) Images (JPG, PNG, GIF)
    C) Fonts (WOFF, TTF)
    D) All of the above

Answer: D) All of the above

---

26. Express and JSON Handling

83. **Which middleware function is used to parse JSON data in Express?** A) `express.json()`

B) `app.json()`
C) `express.body()`
D) `app.use()`

Answer: A) `express.json()`

84. **What does `express.urlencoded()` middleware do in Express?** A) Parses incoming requests with JSON payloads
B) Converts URL-encoded data into a JavaScript object
C) Converts XML data into a JavaScript object
D) Transforms data into URL-encoded format

Answer: B) Converts URL-encoded data into a JavaScript object

---

27. Working with Databases in Node.js

85. **Which of the following is a popular SQL database used with Node.js?** A) MongoDB
B) MySQL
C) Redis
D) Cassandra

Answer: B) MySQL

86. **Which Node.js ORM is commonly used to interact with relational databases like MySQL and PostgreSQL?** A) Sequelize
B) Mongoose
C) Knex.js
D) TypeORM

Answer: A) Sequelize

87. **Which of the following is a key feature of NoSQL databases like MongoDB?** A) They are highly structured
B) They store data in tables
C) They are more flexible in terms of schema design
D) They use SQL to query data

Answer: C) They are more flexible in terms of schema design

---

28. Express API with MongoDB

88. **What is the correct method to connect to MongoDB using Mongoose in Node.js?**
A) `mongoose.db.connect()`
B) `mongoose.connect('mongodb://localhost/dbname')`

C) `mongoose.connectToDB()`
D) `mongoose.startConnection()`

Answer: B) `mongoose.connect('mongodb://localhost/dbname')`

89. **Which of the following is the correct way to define a schema in Mongoose?** A) `mongoose.Schema({name: String, age: Number})`
    B) `mongoose.Schema({name: "String", age: 25})`
    C) `mongoose.Schema("users", {name, age})`
    D) `mongoose.createSchema({name: "String", age: "Number"})`

Answer: A) `mongoose.Schema({name: String, age: Number})`

90. **What method would you use in Mongoose to find all documents in a collection?**
    A) `Model.get()`
    B) `Model.find()`
    C) `Model.query()`
    D) `Model.fetch()`

Answer: B) `Model.find()`

---

29. Authentication with JWT in Node.js

91. **Which of the following is the correct way to create a JWT token in Node.js using `jsonwebtoken`?** A) `jwt.createToken()`
    B) `jwt.encode()`
    C) `jwt.sign(payload, secret)`
    D) `jwt.generateToken()`

Answer: C) `jwt.sign(payload, secret)`

92. **Which method is used to verify the authenticity of a JWT in Node.js?** A) `jwt.verify()`
    B) `jwt.decode()`
    C) `jwt.validate()`
    D) `jwt.check()`

Answer: A) `jwt.verify()`

This brings us closer to completing the desired collection of questions across all the key topics. If you would like more questions for any specific sections, feel free to let me know! I can keep adding more until we hit the target of 200 questions.

4o mini

You said:

30. Client-Side vs Server-Side Rendering (CSR vs SSR)

93. **Which of the following is a benefit of Client-Side Rendering (CSR)?** A) Faster initial page load times
    B) Better SEO optimization
    C) Easier to debug and develop
    D) Reduced load on the server

Answer: C) Easier to debug and develop

94. **Which of the following is a disadvantage of Client-Side Rendering (CSR)?** A) Slower initial page load times due to large JavaScript files
    B) Less interactivity compared to SSR
    C) Inability to render dynamic content
    D) Better SEO performance

Answer: A) Slower initial page load times due to large JavaScript files

95. **Which of the following is a key advantage of Server-Side Rendering (SSR)?** A) Reduces server-side load
    B) Offers faster time-to-interactivity for users
    C) Relies heavily on JavaScript execution in the browser
    D) Easier to implement than CSR

Answer: B) Offers faster time-to-interactivity for users

96. **In which situation would Server-Side Rendering (SSR) be preferred over Client-Side Rendering (CSR)?** A) When SEO and faster content rendering are critical
    B) When the application is purely dynamic and does not need SEO
    C) When client-side JavaScript is not supported
    D) When the application is highly interactive

Answer: A) When SEO and faster content rendering are critical

97. **Which of the following tools or frameworks can be used to implement Server-Side Rendering (SSR) with React?** A) React Native
    B) Next.js
    C) Angular Universal
    D) Vue CLI

Answer: B) Next.js

---

31. Working with Template Engines (EJS, Pug)

98. **What is the primary purpose of a template engine like EJS in a Node.js application?** A) To handle database queries

B) To render dynamic HTML views with embedded JavaScript
C) To manage routing logic
D) To manage user authentication

Answer: B) To render dynamic HTML views with embedded JavaScript

99. **Which of the following is a feature of the EJS template engine?** A) Supports two-way data binding
B) Can directly manipulate the DOM in the browser
C) Allows embedding of JavaScript expressions inside HTML
D) Only works with static HTML files

Answer: C) Allows embedding of JavaScript expressions inside HTML

100. **How do you render a template with data in Express using EJS?** A) `res.sendFile()`
B) `res.render('template', data)`
C) `res.write(data)`
D) `res.send(data)`

Answer: B) `res.render('template', data)`

101. **Which of the following is a feature of Pug (formerly Jade)?** A) It uses indentation to define the HTML structure
B) It relies on explicit closing tags
C) It cannot be used with Node.js
D) It allows inline CSS styles

Answer: A) It uses indentation to define the HTML structure

102. **Which is the correct syntax for including a partial view in an EJS template?** A) `<%- include 'header' %>`
B) `<%= include 'header' %>`
C) `<div> include 'header' </div>`
D) `{{ include 'header' }}`

Answer: A) `<%- include 'header' %>`

---

32. Connecting Node.js to Databases (SQL, NoSQL)

103. **Which of the following is an example of a NoSQL database?** A) MySQL
B) PostgreSQL
C) MongoDB
D) Oracle

Answer: C) MongoDB

104. **What is the default data format used by MongoDB for storing data?** A) JSON
B) BSON (Binary JSON)
C) XML
D) CSV

Answer: B) BSON (Binary JSON)

105. **Which module in Node.js is commonly used to interact with MongoDB?**
A) mongoose
B) knex
C) sequelize
D) sqlite3

Answer: A) mongoose

106. **Which method in Mongoose is used to find a document by its `id`?** A) `Model.findById()`
B) `Model.getById()`
C) `Model.getOne()`
D) `Model.fetch()`

Answer: A) `Model.findById()`

107. **Which of the following is a key feature of MongoDB?** A) SQL-based query language
B) Supports relational tables
C) Schema-less collections
D) Requires a predefined schema

Answer: C) Schema-less collections

108. **Which of the following is a key feature of SQL databases like MySQL?** A) Schema-less design
B) Horizontal scaling
C) Tables with rows and columns
D) Document-oriented structure

Answer: C) Tables with rows and columns

109. **Which of the following is an example of an SQL database?** A) MongoDB
B) MySQL
C) CouchDB
D) Cassandra

Answer: B) MySQL

---

33. Authentication, Authorization, and Sessions

110. **Which of the following is responsible for maintaining user sessions in a web application?** A) Authentication middleware
B) Passport.js
C) express-session
D) Body-parser

Answer: C) express-session

111. **Which of the following is true about authentication?** A) It determines the user’s permissions
B) It checks whether the user has access to a resource
C) It verifies the identity of a user
D) It is the process of authorizing users

Answer: C) It verifies the identity of a user

112. **Which of the following is responsible for determining what actions a user can perform within an application?** A) Authentication
B) Authorization
C) Session management
D) Cookie parsing

Answer: B) Authorization

113. **What is the role of JWT (JSON Web Token) in an authentication system?**
A) It stores session data on the server
B) It encrypts the user password
C) It is used to verify the user's identity and grant access to protected routes
D) It handles database queries

Answer: C) It is used to verify the user's identity and grant access to protected routes

114. **Which of the following is the correct method to generate a JWT token in Node.js using the `jsonwebtoken` package?** A) `jwt.create()`
B) `jwt.sign(payload, secret)`
C) `jwt.encode()`
D) `jwt.newToken()`

Answer: B) `jwt.sign(payload, secret)`

115. **How do you authenticate a user using JWT in Express?** A) By calling `passport.authenticate()`
B) By checking the `Authorization` header for a valid token
C) By setting the session cookie
D) By using `express-session` middleware

Answer: B) By checking the `Authorization` header for a valid token

---

34. Middleware in Express

116. **What is the primary purpose of middleware in Express?** A) To route requests to different endpoints
B) To modify the request and response objects or end the request-response cycle
C) To serve static files
D) To perform database queries

Answer: B) To modify the request and response objects or end the request-response cycle

117. **Which of the following middleware functions is commonly used to parse incoming JSON request bodies?** A) `express.json()`
B) `bodyParser.json()`
C) `express.urlencoded()`
D) `cookieParser()`

Answer: A) `express.json()`

118. **What type of middleware is used to handle errors in Express?** A) Application-level middleware
B) Error-handling middleware
C) Router-level middleware
D) Third-party middleware

Answer: B) Error-handling middleware

119. **Which of the following is true about middleware in Express?** A) Middleware functions execute only once
B) Middleware can only modify the request object
C) Middleware can be chained together to handle complex requests
D) Middleware can only handle GET requests

Answer: C) Middleware can be chained together to handle complex requests

---

35. Working with File Uploads (Multer)

120. **Which of the following is used to handle multipart form-data, typically for file uploads in Node.js?** A) `express-fileupload`
B) `multer`
C) `formidable`
D) `body-parser`

Answer: B) `multer`

121. **Which storage engine in Multer is used for storing files on disk?** A) `memoryStorage`
B) `diskStorage`

C) `fileStorage`
D) `uploadStorage`

Answer: B) `diskStorage`

122. **Which of the following is the correct way to define a file upload limit using Multer?** A) `upload.limit(10MB)`
B) `upload.size('10MB')`
C) `upload.single('file').limit(10MB)`
D) `upload.single('file').maxSize(10)`

Answer: A) `upload.limit(10MB)`

36. Advanced Concepts in Express.js

123. **Which of the following is used to handle multiple HTTP methods (e.g., GET, POST) for the same route in Express?** A) `app.all()`
B) `app.use()`
C) `app.route()`
D) `app.method()`

Answer: C) `app.route()`

124. **What does the `req.body` object in Express represent?** A) The query parameters in the request URL
B) The data sent in a POST or PUT request
C) The headers of the request
D) The path of the requested URL

Answer: B) The data sent in a POST or PUT request

125. **Which of the following is the correct way to redirect a user to another route in Express?** A) `res.redirect('/newpath')`
B) `res.sendRedirect('/newpath')`
C) `res.moveTo('/newpath')`
D) `res.location('/newpath')`

Answer: A) `res.redirect('/newpath')`

126. **In Express, what does the `res.json()` method do?** A) Sends a plain text response
B) Sends a JSON response to the client
C) Converts a JavaScript object into a JSON string
D) Sends an HTML response

Answer: B) Sends a JSON response to the client

127. **Which of the following middleware in Express is used for serving static files like images, CSS, and JavaScript?** A) `express.static()`
B) `express.serve()`
C) `express.files()`
D) `express.staticFiles()`

Answer: A) `express.static()`

---

37. Asynchronous Handling in Node.js

128. **Which of the following functions is used to handle asynchronous code in Node.js without blocking the event loop?** A) `setTimeout()`
B) `setImmediate()`
C) `process.nextTick()`
D) All of the above

Answer: D) All of the above

129. **Which of the following is a method used to handle errors in promises?** A) `.catch()`
B) `.throw()`
C) `.error()`
D) `.onError()`

Answer: A) `.catch()`

130. **What does `async/await` allow you to do in JavaScript?** A) Write synchronous code that waits for asynchronous operations to complete
B) Write asynchronous code that uses callbacks
C) Execute asynchronous operations sequentially
D) Block the execution of a function until an asynchronous task completes

Answer: A) Write synchronous code that waits for asynchronous operations to complete

131. **Which of the following functions would you use in Node.js to read a file asynchronously?** A) `fs.readFileSync()`
B) `fs.readFile()`
C) `fs.read()`
D) `fs.readAsync()`

Answer: B) `fs.readFile()`

132. **Which of the following methods is used to handle errors in a Node.js asynchronous operation?** A) `try-catch` block
B) `setTimeout()`
C) Using a callback function
D) `.catch()` on Promises

Answer: D) `.catch()` on Promises

---

38. Authentication & Authorization

133. **Which of the following is the best way to store user credentials securely in a Node.js application?** A) Storing passwords in plain text in the database
B) Using `bcrypt` to hash passwords before storing them
C) Storing passwords in cookies
D) Storing passwords in a session

Answer: B) Using `bcrypt` to hash passwords before storing them

134. **Which Node.js package is commonly used for authenticating users via a social login (Google, Facebook)?** A) `passport`
B) `jwt-simple`
C) `cookie-parser`
D) `express-session`

Answer: A) `passport`

135. **What is the main purpose of JWT (JSON Web Token) in user authentication?** A) To send the user's password over the network
B) To generate a session ID that is stored on the server
C) To securely transfer information between parties
D) To hash the user's password for storage

Answer: C) To securely transfer information between parties

136. **How can you ensure that a route is protected and only accessible to authenticated users in Express?** A) By using `app.use()` middleware to check for a valid session or JWT token
B) By calling `next()` in every route handler
C) By checking the IP address of the client
D) By enforcing CORS (Cross-Origin Resource Sharing)

Answer: A) By using `app.use()` middleware to check for a valid session or JWT token

137. **Which of the following is the correct way to create a hashed password using `bcrypt` in Node.js?** A) `bcrypt.hashSync('password', 10)`
B) `bcrypt.hash('password', 10)`
C) `bcrypt.encrypt('password')`
D) `bcrypt.encryptSync('password', 10)`

Answer: A) `bcrypt.hashSync('password', 10)`

---

39. File Uploads with Multer

138. **In Multer, which of the following is used to define where files will be stored?** A) `storage`
     B) `destination`
     C) `uploadTo`
     D) `store`

Answer: A) `storage`

139. **Which of the following is a valid method to specify limits on file size in Multer?** A) `upload.size(10MB)`
     B) `upload.fileSize(10MB)`
     C) `upload.limit({ fileSize: 10MB })`
     D) `upload.setSize(10MB)`

Answer: C) `upload.limit({ fileSize: 10MB })`

140. **Which Multer method is used to handle a single file upload?** A) `upload.single('file')`
     B) `upload.multi('file')`
     C) `upload.uploadOne('file')`
     D) `upload.file()`

Answer: A) `upload.single('file')`

141. **Which property in the Multer `request` object holds the uploaded file information?** A) `req.body`
     B) `req.files`
     C) `req.file`
     D) `req.data`

Answer: C) `req.file`

142. **Which of the following is the correct method to configure the Multer storage engine for saving files to disk?** A) `storage: multer.diskStorage()`
     B) `storage: multer.memoryStorage()`
     C) `storage: multer.createStorage()`
     D) `storage: multer.config()`

Answer: A) `storage: multer.diskStorage()`

---

40. Using NPM (Node Package Manager)

143. **Which of the following commands initializes a new Node.js project and creates a `package.json` file?** A) `npm start`
     B) `npm init`

C) `npm install`
D) `npm run`

Answer: B) `npm init`

144. **How can you install a specific version of a package using npm?** A) `npm install package@<version>`
B) `npm install package<version>`
C) `npm install -version <version>`
D) `npm install <version>`

Answer: A) `npm install package@<version>`

145. **Which command can be used to install all dependencies listed in `package.json`?** A) `npm update`
B) `npm install`
C) `npm start`
D) `npm dependencies`

Answer: B) `npm install`

146. **Which of the following is the correct way to add a package as a development dependency in npm?** A) `npm add <package> --dev`
B) `npm install <package> --dev`
C) `npm install <package> --save-dev`
D) `npm install <package> --prod`

Answer: C) `npm install <package> --save-dev`

147. **Which command is used to uninstall a package from your Node.js project using npm?** A) `npm uninstall <package>`
B) `npm remove <package>`
C) `npm delete <package>`
D) `npm uninstall <package> --global`

Answer: A) `npm uninstall <package>`

---

41. Working with RESTful APIs in Node.js

148. **In a RESTful API, what does the HTTP GET method typically do?** A) Submits data to be processed
B) Retrieves data from the server
C) Deletes a resource
D) Updates a resource

Answer: B) Retrieves data from the server

149. **Which of the following is an example of a RESTful URL path?** A) `GET /api/user/123`
B) `POST /api/user?id=123`
C) `GET /api/user?id=123`
D) `DELETE /api/user`

Answer: A) `GET /api/user/123`

150. **In Express, which method is used to handle a PUT request?** A) `app.put()`
B) `app.post()`
C) `app.get()`
D) `app.delete()`

Answer: A) `app.put()`

42. Error Handling in Express

151. **Which of the following is the correct way to handle errors in Express?** A) By using a try-catch block in each route
B) By using `next()` to pass errors to the error-handling middleware
C) By calling `res.error()`
D) By logging errors to the console

Answer: B) By using `next()` to pass errors to the error-handling middleware

152. **Which type of middleware function in Express is used to handle errors?**
A) Application-level middleware
B) Router-level middleware
C) Error-handling middleware
D) Request middleware

Answer: C) Error-handling middleware

153. **In Express, what will happen if an error is passed to `next()` in a route handler?** A) The request will be completed successfully
B) The error will be logged in the server
C) The request will be forwarded to the error-handling middleware
D) The error will be silently ignored

Answer: C) The request will be forwarded to the error-handling middleware

154. **Which of the following is the correct syntax for an error-handling middleware function in Express?** A) `app.use(errorHandler(req, res))`
B) `app.use((err, req, res, next) => {...})`
C) `app.use((err, next) => {...})`
D) `app.use((err, res, req) => {...})`

Answer: B) `app.use((err, req, res, next) => {...})`

155. **In Express, how do you respond to the client when an error occurs?** A) `res.sendError()`
    B) `res.status(500).send('Internal Server Error')`
    C) `res.sendMessage('Error')`
    D) `res.error()`

Answer: B) `res.status(500).send('Internal Server Error')`

---

43. Advanced Node.js Concepts

156. **Which of the following modules is part of the Node.js core and allows you to handle HTTP requests and responses?** A) `express`
    B) `http`
    C) `mongoose`
    D) `socket.io`

Answer: B) `http`

157. **What is the purpose of the `cluster` module in Node.js?** A) It helps to create a server to handle HTTP requests
    B) It provides a way to fork multiple processes to utilize multi-core systems
    C) It handles file uploads in Node.js
    D) It handles authentication in Node.js

Answer: B) It provides a way to fork multiple processes to utilize multi-core systems

158. **What is the default behavior of the Node.js event loop?** A) It processes synchronous operations first and asynchronous operations second
    B) It processes asynchronous operations first and synchronous operations second
    C) It processes both asynchronously
    D) It only processes synchronous operations

Answer: A) It processes synchronous operations first and asynchronous operations second

159. **Which of the following is used to create streams in Node.js?** A) `stream` module
    B) `buffer` module
    C) `http` module
    D) `express` module

Answer: A) `stream` module

160. **What type of I/O model does Node.js follow?** A) Synchronous
    B) Asynchronous

C) Blocking
D) Polling

Answer: B) Asynchronous

---

44. Connecting Node.js to MongoDB

161. **What method would you use to connect to a MongoDB database in Node.js using Mongoose?** A) `mongoose.connect()`
B) `mongoose.connectDB()`
C) `mongoose.open()`
D) `mongoose.link()`

Answer: A) `mongoose.connect()`

162. **Which of the following is used to define a model in Mongoose?** A) `mongoose.model()`
B) `mongoose.Schema()`
C) `mongoose.create()`
D) `mongoose.connect()`

Answer: A) `mongoose.model()`

163. **Which of the following is the correct way to create a new document using Mongoose?** A) `Model.save()`
B) `Model.create()`
C) `Model.insert()`
D) `Model.new()`

Answer: B) `Model.create()`

164. **Which method in Mongoose is used to update a document?** A) `Model.update()`
B) `Model.modify()`
C) `Model.set()`
D) `Model.updateOne()`

Answer: D) `Model.updateOne()`

165. **What type of database is MongoDB?** A) Relational database
B) Key-value store
C) NoSQL document-oriented database
D) In-memory database

Answer: C) NoSQL document-oriented database

---

45. Authentication & Authorization with JWT

166. **Which of the following JWT sections contains the claims or data about the user?** A) Header
B) Payload
C) Signature
D) Header and Signature

Answer: B) Payload

167. **Which of the following is a typical use case for JSON Web Tokens (JWT)?** A) Storing the user's password securely
B) Sending authentication credentials between client and server
C) Encrypting sensitive data in the database
D) Creating session cookies for a client

Answer: B) Sending authentication credentials between client and server

168. **Which of the following HTTP headers is typically used to send a JWT from a client to a server?** A) `Authorization`
B) `Content-Type`
C) `X-Auth-Token`
D) `Cookie`

Answer: A) `Authorization`

169. **What does JWT's "HS256" algorithm refer to?** A) The method used to sign the token
B) The encryption standard used in JWT
C) The number of claims in the token
D) The length of the JWT token

Answer: A) The method used to sign the token

170. **How would you decode a JWT token in Node.js using the `jsonwebtoken` package?** A) `jwt.verify(token, secret)`
B) `jwt.decode(token)`
C) `jwt.decode(token, secret)`
D) `jwt.parse(token)`

Answer: B) `jwt.decode(token)`

---

46. Static File Handling in Express

171. **Which of the following middleware is used to serve static files in Express?**
A) `express.static()`
B) `express.serveStatic()`

C) `express.fileServe()`
D) `express.sendFile()`

Answer: A) `express.static()`

172. **Which of the following is the correct way to serve static files from the "public" directory in Express?** A) `app.use(express.static('public'))`
B) `app.use(express.fileServe('public'))`
C) `app.static('public')`
D) `app.serve('public')`

Answer: A) `app.use(express.static('public'))`

173. **Which Express method would you use to send a static HTML file in response to a GET request?** A) `res.sendFile()`
B) `res.send()`
C) `res.render()`
D) `res.view()`

Answer: A) `res.sendFile()`

---

47. Express Middleware and Routing

174. **Which of the following is the correct way to define a route handler for a GET request in Express?** A) `app.route('/path').get()`
B) `app.get('/path')`
C) `app.use('/path', 'get')`
D) `app.handle('/path', 'GET')`

Answer: B) `app.get('/path')`

175. **What is the purpose of `app.use()` in Express?** A) It is used to define route handlers for HTTP methods
B) It is used to declare middleware functions that run for every incoming request
C) It is used to start the HTTP server
D) It is used to connect to a database

Answer: B) It is used to declare middleware functions that run for every incoming request

176. **What does `next()` do in an Express middleware function?** A) It terminates the current middleware and sends a response to the client
B) It passes the request to the next middleware or route handler in the stack
C) It logs the request and continues to the next request
D) It processes the request synchronously

Answer: B) It passes the request to the next middleware or route handler in the stack

177. **Which of the following is used to handle route parameters in Express?** A) `req.body`
B) `req.query`
C) `req.params`
D) `req.path`

Answer: C) `req.params`

---

48. Client-Side Rendering vs Server-Side Rendering

178. **What is the primary difference between client-side rendering (CSR) and server-side rendering (SSR)?** A) CSR sends HTML content from the server to the client, while SSR sends JavaScript content
B) CSR executes JavaScript on the client-side to render the content, while SSR generates HTML on the server and sends it to the client
C) CSR is only possible in React applications, while SSR is used in Vue.js applications
D) CSR uses fewer resources on the server, while SSR uses fewer resources on the client

Answer: B) CSR executes JavaScript on the client-side to render the content, while SSR generates HTML on the server and sends it to the client

179. **Which of the following is a key advantage of SSR over CSR?** A) SSR tends to have better SEO performance since the HTML content is fully rendered on the server
B) SSR has faster page loading times because the server sends raw JavaScript
C) SSR is less resource-intensive for the client compared to CSR
D) SSR works only with JavaScript frameworks like React

Answer: A) SSR tends to have better SEO performance since the HTML content is fully rendered on the server

49. Template Engines in Node.js

180. **Which of the following is a popular template engine used with Express.js?**
A) Pug (formerly Jade)
B) Handlebars
C) EJS
D) All of the above

Answer: D) All of the above

181. **Which of the following is the correct way to set the view engine to EJS in Express?** A) `app.set('view', 'ejs')`
B) `app.set('view engine', 'ejs')`
C) `app.set('viewTemplate', 'ejs')`
D) `app.use('ejs')`

Answer: B) `app.set('view engine', 'ejs')`

182. **Which EJS syntax is used to include another EJS template inside the current template?** A) `<% include file.ejs %>`
    B) `<% import file.ejs %>`
    C) `<% require('file.ejs') %>`
    D) `<% include('file.ejs') %>`

Answer: A) `<% include file.ejs %>`

183. **What does the EJS `<%= %>` tag do?** A) It outputs raw JavaScript code
    B) It outputs the result of a JavaScript expression to the HTML
    C) It is used to include a template
    D) It marks a section of HTML to be ignored

Answer: B) It outputs the result of a JavaScript expression to the HTML

184. **How can you pass data from a route handler to an EJS view in Express?**
    A) `res.send(data)`
    B) `res.render('view', { data })`
    C) `res.json(data)`
    D) `res.view('view', data)`

Answer: B) `res.render('view', { data })`

50. Session Management in Node.js

185. **Which of the following is commonly used for session management in Express?** A) `express-session`
    B) `session-store`
    C) `cookie-parser`
    D) `passport-session`

Answer: A) `express-session`

186. **Which of the following is the correct way to initialize session middleware in an Express app?** A) `app.use(express.session())`
    B) `app.use(session())`
    C) `app.use(session({ secret: 'your-secret-key' }))`
    D) `app.use(cookieParser())`

Answer: C) `app.use(session({ secret: 'your-secret-key' }))`

187. **Which of the following is a method used to destroy a session in Express?**
    A) `req.session.destroy()`
    B) `req.session.delete()`

C) `session.remove()`
D) `session.destroy()`

Answer: A) `req.session.destroy()`

188. **What is the purpose of the `cookie-parser` middleware in Express?** A) It parses cookies in the request and makes them accessible via `req.cookies`
B) It encrypts the cookies before sending them to the client
C) It manages session cookies for the application
D) It handles file uploads as cookies

Answer: A) It parses cookies in the request and makes them accessible via `req.cookies`

189. **Which of the following is a way to manage user authentication sessions using JWT?** A) Store the JWT in the session
B) Store the JWT in a cookie or local storage
C) Store the JWT in the database
D) Store the JWT in the server's memory

Answer: B) Store the JWT in a cookie or local storage

---

51. File Handling in Node.js

190. **Which Node.js module is used for interacting with the file system?** A) `fs`
B) `fileSystem`
C) `path`
D) `file-system`

Answer: A) `fs`

191. **Which of the following is the correct method to read a file asynchronously in Node.js?** A) `fs.readFileSync()`
B) `fs.readFile()`
C) `fs.read()`
D) `fs.open()`

Answer: B) `fs.readFile()`

192. **Which of the following methods can be used to write to a file asynchronously in Node.js?** A) `fs.writeFileSync()`
B) `fs.writeFile()`
C) `fs.appendFileSync()`
D) `fs.appendFile()`

Answer: B) `fs.writeFile()`

193. **Which method in the `fs` module would you use to check if a file exists?** A) `fs.existsSync()`
B) `fs.checkFile()`
C) `fs.isExist()`
D) `fs.exists()`

Answer: A) `fs.existsSync()`

194. **How would you stream a large file in Node.js to avoid blocking the event loop?** A) Using `fs.readFile()`
B) Using `fs.createReadStream()`
C) Using `fs.writeFileSync()`
D) Using `fs.appendFile()`

Answer: B) Using `fs.createReadStream()`

52. MongoDB with Node.js (Mongoose)

195. **Which method in Mongoose is used to find all documents in a collection?**
A) `Model.find()`
B) `Model.getAll()`
C) `Model.get()`
D) `Model.queryAll()`

Answer: A) `Model.find()`

196. **How do you connect to a MongoDB database using Mongoose in Node.js?**
A) `mongoose.connect('mongodb://localhost/mydb')`
B) `mongoose.db('mongodb://localhost/mydb')`
C) `mongoose.connectDB('mongodb://localhost/mydb')`
D) `mongoose.connectTo('mongodb://localhost/mydb')`

Answer: A) `mongoose.connect('mongodb://localhost/mydb')`

197. **What does `Model.findOne()` do in Mongoose?** A) It retrieves multiple documents that match the query
B) It retrieves a single document that matches the query
C) It deletes a document from the collection
D) It updates a document in the collection

Answer: B) It retrieves a single document that matches the query

198. **What is the purpose of the `Schema` in Mongoose?** A) It defines the structure of the MongoDB documents for a specific collection
B) It is used to create queries to find documents
C) It maps the documents to a specific view in an application
D) It stores the database connection information

Answer: A) It defines the structure of the MongoDB documents for a specific collection

199. **Which method is used to save a document in a MongoDB collection using Mongoose?** A) `Model.save()`
B) `Model.insert()`
C) `Model.create()`
D) `Model.update()`

Answer: A) `Model.save()`

---

53. Performance Optimization in Node.js

200. **Which of the following is an advantage of using Node.js for building scalable applications?** A) It is single-threaded, allowing it to process many requests simultaneously without blocking
B) It is multi-threaded, allowing it to handle multiple requests in parallel
C) It uses a client-server model that reduces network latency
D) It uses a binary protocol for faster communication

Answer: A) It is single-threaded, allowing it to process many requests simultaneously without blocking

201. **Which of the following is a Node.js tool that helps with debugging and performance profiling?** A) `node-inspect`
B) `node-debugger`
C) `node-profile`
D) `node-track`

Answer: A) `node-inspect`

202. **What is the primary reason Node.js is known for being efficient in handling I/O operations?** A) Node.js runs on a single thread using non-blocking I/O operations
B) Node.js executes all tasks synchronously
C) Node.js performs caching of database queries
D) Node.js uses a multi-threaded event loop

Answer: A) Node.js runs on a single thread using non-blocking I/O operations

203. **Which of the following is used to minimize memory consumption and avoid memory leaks in Node.js applications?** A) Event-driven architecture
B) Using callback functions
C) Using streams to handle large datasets
D) Using setInterval() for background tasks

Answer: C) Using streams to handle large datasets

204. **Which of the following Node.js modules can be used to measure the performance and resource usage of a Node.js application?** A) `os`
B) `v8`
C) `process`
D) All of the above

Answer: D) All of the above

54. File Uploads in Node.js

205. **Which module is commonly used for handling file uploads in Node.js?** A) `multer`
B) `formidable`
C) `busboy`
D) All of the above

Answer: D) All of the above

206. **How do you use `multer` to handle file uploads in Express?** A) `app.use(multer())`
B) `app.post('/upload', multer())`
C) `app.use(multer({ dest: 'uploads/' }))`
D) `app.use(multer({ dest: 'uploads/' }).single('file'))`

Answer: D) `app.use(multer({ dest: 'uploads/' }).single('file'))`

207. **Which method in `multer` is used to store files on disk?** A) `diskStorage()`
B) `memoryStorage()`
C) `uploadFile()`
D) `storeFile()`

Answer: A) `diskStorage()`

208. **What is the role of `req.file` in the file upload process using `multer`?** A) It contains information about the uploaded file, such as the file name and path
B) It contains the entire file content in the buffer format
C) It is a method used to retrieve the uploaded file
D) It contains the MIME type of the uploaded file

Answer: A) It contains information about the uploaded file, such as the file name and path

209. **What is the maximum file size allowed by `multer` by default?** A) 10MB
B) 1MB
C) Unlimited
D) 2MB

Answer: C) Unlimited (But you can set a custom limit)

---

55. Debugging in Node.js

210. **Which Node.js debugger tool is built into Node.js and can be used for interactive debugging?** A) `node-inspect`
    B) `node-debug`
    C) `node-logger`
    D) `node-debugger`

Answer: A) `node-inspect`

211. **Which of the following commands can be used to start debugging a Node.js application?** A) `node debug app.js`
    B) `node --inspect app.js`
    C) `node --debug app.js`
    D) `node --log app.js`

Answer: B) `node --inspect app.js`

212. **What is the purpose of `console.log()` in Node.js debugging?** A) To output debug logs to the console
    B) To pause the execution of the script
    C) To trigger an exception in the code
    D) To check the value of a variable

Answer: A) To output debug logs to the console

213. **In Node.js, which tool can be used to inspect memory leaks and performance issues?** A) `heapdump`
    B) `node-inspect`
    C) `v8-profiler`
    D) Both A and C

Answer: D) Both A and C

214. **Which of the following can you use to get stack traces in Node.js to help debug asynchronous code?** A) `process.on('uncaughtException', callback)`
    B) `console.trace()`
    C) `node-inspect`
    D) `setTimeout()`

Answer: B) `console.trace()`

---

56. Security in Node.js Applications

215. **Which of the following is used to prevent Cross-Site Scripting (XSS) attacks in Node.js?** A) Use `helmet` middleware
B) Use `express-session` middleware
C) Sanitize user input
D) All of the above

Answer: D) All of the above

216. **Which package is commonly used to secure HTTP headers and prevent common attacks in Node.js applications?** A) `helmet`
B) `cors`
C) `express-rate-limit`
D) `body-parser`

Answer: A) `helmet`

217. **What is Cross-Origin Resource Sharing (CORS)?** A) A way to enable resource sharing across different servers
B) A middleware to parse incoming requests
C) A method of encrypting API keys
D) A Node.js library for file handling

Answer: A) A way to enable resource sharing across different servers

218. **Which of the following best describes SQL Injection attacks?** A) Attacks that modify the client-side UI of a web page
B) Attacks that allow an attacker to inject malicious code into the database
C) Attacks that prevent the server from sending responses
D) Attacks that exploit client-side scripts for data theft

Answer: B) Attacks that allow an attacker to inject malicious code into the database

219. **How can you prevent SQL Injection attacks in a Node.js application?** A) Use prepared statements or parameterized queries
B) Sanitize and validate user input
C) Avoid concatenating SQL queries directly with user data
D) All of the above

Answer: D) All of the above

---

57. Advanced Node.js Features

220. **What is the role of the `cluster` module in Node.js?** A) It handles multiple threads of execution on different machines
B) It allows Node.js to be run on multiple CPU cores by creating child processes
C) It manages database connections for Node.js applications
D) It helps to manage in-memory caches

Answer: B) It allows Node.js to be run on multiple CPU cores by creating child processes

221. **Which of the following statements about the `async` and `await` keywords in Node.js is correct?** A) `async` makes a function return a Promise, and `await` pauses the execution until the Promise resolves
B) `async` pauses the execution, and `await` returns a Promise
C) Both `async` and `await` are used to handle callbacks
D) `async` and `await` are not supported in Node.js

Answer: A) `async` makes a function return a Promise, and `await` pauses the execution until the Promise resolves

222. **Which of the following is true about `Promise.all()` in Node.js?** A) It allows multiple promises to be executed sequentially
B) It resolves when all the promises in the array are fulfilled
C) It resolves as soon as the first promise in the array resolves
D) It is used to handle errors in promises

Answer: B) It resolves when all the promises in the array are fulfilled

223. **Which of the following is the correct way to handle an error in an `async` function in Node.js?** A) Using a `try-catch` block
B) Using the `.catch()` method on the promise
C) Using the `Promise.reject()` method
D) Both A and B

Answer: D) Both A and B

224. **Which of the following modules is commonly used to work with streams in Node.js?** A) `stream`
B) `http`
C) `express`
D) `socket.io`

Answer: A) `stream`

---

58. Optimizing Express Applications

225. **Which of the following strategies can be used to optimize performance in an Express application?** A) Enable caching of static assets
B) Minimize the use of synchronous code
C) Use gzip compression for responses
D) All of the above

Answer: D) All of the above

226. **What is the purpose of the `compression` middleware in Express?** A) It compresses responses to reduce the size of the data sent to the client
B) It ensures requests are processed in parallel
C) It encrypts the response body
D) It caches the server's response

Answer: A) It compresses responses to reduce the size of the data sent to the client

227. **Which of the following is a common method for managing static assets in Express for better performance?** A) Use `express.static()` middleware
B) Store static files in the database
C) Compress static files using Gzip
D) Serve static files directly from the client

Answer: A) Use `express.static()` middleware

228. **What is the role of `app.set('trust proxy', true)` in Express?** A) It allows Express to handle HTTPS connections
B) It enables the use of secure cookies and tracks the originating IP address when the app is behind a proxy
C) It enables the use of the session middleware
D) It caches responses to improve performance

Answer: B) It enables the use of secure cookies and tracks the originating IP address when the app is behind a proxy

229. **Which of the following can help improve the performance of an Express application?** A) Use `app.use()` for common middleware
B) Minimize the number of middleware functions executed per request
C) Use asynchronous handlers for I/O-bound operations
D) All of the above

Answer: D) All of the above