100+ Most Asked Node.js QnA

Made By:

CommonJobs

1. **What is Node.js?**
   Node.js is an open-source JavaScript runtime environment built on Chrome's V8 JavaScript engine. It allows developers to execute JavaScript code on the server-side, enabling server-side scripting and the development of scalable network applications.
2. **What are the key features of Node.js?**
   Some key features of Node.js include:
3. Asynchronous and event-driven: It uses an event-driven, non-blocking I/O model, making it efficient and scalable.
4. Single-threaded: It employs a single-threaded event loop to handle concurrent requests efficiently.
5. NPM (Node Package Manager): It provides a vast collection of reusable packages and modules.
6. Cross-platform: Node.js is compatible with multiple operating systems.
7. **What is NPM?**
   NPM (Node Package Manager) is the default package manager for Node.js. It allows developers to easily install, manage, and share reusable JavaScript packages/modules.
8. **How do you install packages in Node.js?**
   To install packages in Node.js, you can use the npm install command followed by the package name. For example, npm install express installs the Express framework.
9. **Explain the concept of the event loop in Node.js.**
   The event loop is a key feature of Node.js that allows it to handle concurrent requests efficiently. It constantly listens for events, such as I/O operations or timers, and executes the associated callback functions when an event occurs. This non-blocking approach ensures that the server can handle multiple requests without getting blocked.
10. **What is a callback function in Node.js?**
    A callback function is a function that is passed as an argument to another function and is executed later when a specific event occurs. In Node.js, callbacks are widely used for handling asynchronous operations, such as reading files or making HTTP requests.
11. **What is the purpose of the require function in Node.js?**
    The require function is used to include modules in Node.js. It allows you to load built-in modules or external modules installed via NPM into your application.

12. **Explain the concept of middleware in Express.js.**
Middleware functions in Express.js are functions that have access to the request and response objects. They can modify the request/response, execute additional code, and pass control to the next middleware function in the stack. Middleware is commonly used for tasks such as authentication, logging, and error handling.
13. **What is the difference between process.nextTick and setImmediate in Node.js?**
14. process.nextTick queues a function to be executed on the next iteration of the event loop. It has higher priority than other asynchronous operations.
15. setImmediate queues a function to be executed in the next iteration of the event loop but with lower priority than process.nextTick.
16. **What is the purpose of the fs module in Node.js?**
The fs module in Node.js provides file system-related functionality, such as reading and writing files, creating directories, and manipulating file paths.
17. **How can you handle errors in Node.js?**
In Node.js, errors can be handled using try-catch blocks or by using error handling middleware. The try-catch block is used for synchronous code, while error handling middleware is used for asynchronous operations.
18. **What is the purpose of the Buffer class in Node.js?**
The Buffer class in Node.js provides a way to handle binary data. It allows you to read from and write to binary streams and manipulate raw binary data.
19. **What is a stream in Node.js?**
A stream in Node.js is a mechanismfor handling continuous data flow, both readable and writable. It allows you to process large amounts of data in chunks, making it memory-efficient and suitable for handling data from sources like files, network connections, or HTTP requests.
20. **What are the different types of streams in Node.js?**
There are four types of streams in Node.js:
21. Readable streams: Used for reading data from a source.
22. Writable streams: Used for writing data to a destination.
23. Duplex streams: Both readable and writable, allowing data to be read from and written to simultaneously.
24. Transform streams: A type of duplex stream that can modify or transform the data as it passes through.
25. **What is the purpose of the crypto module in Node.js?**
The crypto module in Node.js provides cryptographic functionality, such as generating hashes, creating digital signatures, encrypting and decrypting data, and performing secure communication over networks.
26. **Explain the concept of clustering in Node.js.**
Clustering in Node.js allows you to create multiple worker processes (child processes) that can share the same server port. It helps to utilize multiple CPU cores and improve the overall performance and scalability of Node.js applications.
27. **What is the difference between fork and spawn methods in Node.js?**
28. The fork method is used to create child processes that run Node.js modules. It sets up inter-process communication (IPC) channels between the parent and child

processes.

29. The spawn method is used to create child processes and execute external commands. It does not set up IPC channels by default.
30. **What is the purpose of the cluster module in Node.js?**
    The cluster module in Node.js provides an easy way to implement clustering. It allows you to create a cluster of Node.js processes that can share the same server port and distribute the incoming requests among the worker processes.
31. **What is Express.js?**
    Express.js is a popular web application framework for Node.js. It provides a simple and flexible way to build web applications and APIs, handling routes, middleware, and other web-related functionalities.
32. **How do you create a basic server using Express.js?**
    To create a basic server using Express.js, you need to:
33. Import the Express module (const express = require('express');).
34. Create an instance of the Express application (const app = express();).
35. Define routes and their corresponding handlers (app.get('/', (req, res) => { /* Handle request */ });).
36. Start the server by listening on a specific port (app.listen(3000, () => { console.log('Server started on port 3000'); });).
37. **What is middleware in Express.js?**
    Middleware in Express.js are functions that have access to the request and response objects. They can perform tasks like modifying request/response objects, executing additional code, or passing control to the next middleware function. Middleware functions are executed in a sequential order as defined.
38. **What is the purpose of the body-parser middleware in Express.js?**
    The body-parser middleware in Express.js is used to parse the body of incoming requests. It extracts data from the request body and makes it available in req.body for further processing.
39. **How do you handle routing in Express.js?**
    In Express.js, you can define routes using the app.get(), app.post(), app.put(), app.delete(), and other methods provided by the Express application. Each route maps to a specific URL and specifies a callback function that handles the request and generates the response.
40. **Explain the concept of template engines in Express.js.**
    Template engines in Express.jsare used to dynamically generate HTML pages by combining data with predefined HTML templates. Express.js supports various template engines like EJS, Pug (formerly Jade), Handlebars, etc. These engines allow you to embed dynamic data within the HTML template, making it easier to generate dynamic web pages.
41. **What is the difference between app.use() and app.METHOD() in Express.js?**
42. app.use() is a middleware function that is executed for every incoming request, regardless of the HTTP method. It is commonly used for tasks like setting up middleware, defining routes, and handling error middleware.

43. app.METHOD() (e.g., app.get(), app.post(), app.put()) is used to define route-specific middleware that is executed only for the specified HTTP method and route.
44. **What is the purpose of the dotenv module in Node.js?**
    The dotenv module in Node.js allows you to load environment variables from a .env file into the process.env object. It provides an easy way to manage and access configuration settings for your Node.js application.
45. **What is the purpose of the async module in Node.js?**
    The async module in Node.js provides a powerful set of utility functions for handling asynchronous operations. It allows you to control the flow of asynchronous code using functions like async.waterfall(), async.parallel(), async.series(), etc.
46. **What is the purpose of the mongoose module in Node.js?**
    The mongoose module is an Object-Data Modeling (ODM) library for MongoDB and Node.js. It provides an elegant and intuitive way to interact with MongoDB databases, defining schemas, models, and performing database operations.
47. **Explain the concept of middleware in the context of error handling in Express.js.**
    In the context of error handling in Express.js, middleware functions are used to handle errors that occur during the processing of requests. Error handling middleware is defined with four parameters (err, req, res, next) and is executed when an error occurs. It can handle the error, log it, and send an appropriate response to the client.
48. **What is the purpose of the cookie-parser middleware in Express.js?**
    The cookie-parser middleware in Express.js is used to parse and handle HTTP cookies. It allows you to read and write cookies in the request/response objects, enabling cookie-based session management and other cookie-related operations.
49. **What is the purpose of the express-session middleware in Express.js?**
    The express-session middleware in Express.js provides session management capabilities. It allows you to create and manage user sessions, store session data, and handle session-related operations like session-based authentication and authorization.
50. **What is the purpose of the socket.io library in Node.js?**
    The socket.io library in Node.js enables real-time bidirectional communication between the client and the server. It provides a set of APIs for building WebSocket-based applications, allowing for real-time data exchange and event-driven communication.
51. **What is the purpose of the jsonwebtoken library in Node.js?**
    The jsonwebtoken library in Node.js is used for generating and verifying JSON Web Tokens (JWTs). JWTs are used for authentication and authorization purposes, allowing secure transmission of claims between parties.
52. **Explain the concept of Promises in Node.js.**
    Promises in Node.js provide a cleaner way to handle asynchronous operations. They represent the eventual completion (or failure) of an asynchronous operation and allow you to chain multiple asynchronous operations together, making code more readable and avoiding callback hell.
53. **What are the different states of a Promise inNode.js?**
    A Promise in Node.js can be in one of three states:

54. Pending: The initial state of a Promise. It indicates that the asynchronous operation is still ongoing and has not yet been fulfilled or rejected.

55. Fulfilled: The state of a Promise when the asynchronous operation has completed successfully, and the associated value (result) is available.

56. Rejected: The state of a Promise when the asynchronous operation has encountered an error or failure, and the associated reason (error) is available.

57. **How do you handle errors in Promises?**
In Promises, you can handle errors using the catch method or by chaining the catch method after the then method. This allows you to handle any errors that occur during the asynchronous operation and perform appropriate error handling or fallback actions.

58. **What is the purpose of the async/await feature in Node.js?**
The async/await feature in Node.js provides a more synchronous-style way to write asynchronous code. It allows you to write asynchronous operations using synchronous-like syntax, making it easier to read, write, and maintain asynchronous code.

59. **What is the purpose of the EventEmitter class in Node.js?**
The EventEmitter class in Node.js provides an implementation of the publisher-subscriber pattern. It allows objects to emit named events and register listeners that respond to those events, enabling efficient communication and decoupling between different parts of the application.

60. **What is the purpose of the process object in Node.js?**
The process object in Node.js provides information and control over the current Node.js process. It allows you to access command-line arguments, environment variables, exit the process, listen for process events, and more.

61. **How do you handle file uploads in Node.js?**
To handle file uploads in Node.js, you can use middleware like multer. Multer is a popular middleware that handles multipart/form-data, allowing you to process file uploads in your Express.js application.

62. **What is the purpose of the os module in Node.js?**
The os module in Node.js provides a set of utility methods for interacting with the operating system. It allows you to access information about the system's CPUs, memory, network interfaces, and more.

63. **How do you implement caching in Node.js?**
Caching in Node.js can be implemented using various techniques, such as in-memory caching, using caching libraries like Redis or Memcached, or utilizing HTTP caching headers like Cache-Control and ETag. Caching helps improve performance by storing frequently accessed data and reducing the load on the server.

64. **What is the purpose of the child_process module in Node.js?**
The child_process module in Node.js allows you to create and control child

for debugging, error handling, and other common tasks. It includes functions like promisify for converting callback-based functions into Promises, inherits for prototypal inheritance, and more.

66. **What are the differences between setTimeout and setImmediate in Node.js?**
67. setTimeout schedules a function to be executed after a specified delay (in milliseconds). It adds the callback to the timers phase of the event loop.
68. setImmediate schedules a function to be executed in the next iteration of the event loop, immediately after the I/O callbacks phase.
69. **What is the purposeof the URL module in Node.js?**
    The URL module in Node.js provides utilities for working with URLs. It allows you to parse, manipulate, and construct URLs, extract different components (e.g., hostname, path, query parameters), and perform URL-related operations.
70. **What is the purpose of the cluster module in Node.js?**
    The cluster module in Node.js allows you to create a cluster of Node.js processes to take advantage of multiple CPU cores. It enables load balancing and improves the scalability and performance of Node.js applications by distributing the workload among multiple worker processes.
71. **What is the purpose of the http module in Node.js?**
    The http module in Node.js provides functionality for creating HTTP servers and making HTTP requests. It allows you to handle incoming HTTP requests, send HTTP responses, and interact with HTTP servers and clients.
72. **What is the purpose of the https module in Node.js?**
    The https module in Node.js extends the http module and provides functionality for creating HTTPS servers and making secure HTTPS requests. It allows you to handle secure HTTP connections using SSL/TLS certificates.
73. **How can you handle concurrent requests in Node.js?**
    In Node.js, concurrent requests can be handled efficiently due to its asynchronous and non-blocking nature. By using event-driven programming, callbacks, promises, or async/await, you can handle multiple requests concurrently without blocking the execution of other requests.
74. **What is the purpose of the net module in Node.js?**
    The net module in Node.js provides functionality for creating TCP servers and clients. It allows you to create TCP connections, send and receive data over TCP sockets, and build TCP-based networking applications.
75. **What is the purpose of the dns module in Node.js?**
    The dns module in Node.js provides methods for DNS (Domain Name System) operations. It allows you to perform DNS lookups, resolve hostnames to IP addresses, perform reverse DNS lookups, and more.
76. **What is the purpose of the url module in Node.js?**
    The url module in Node.js provides methods for URL resolution and parsing. It allows you to parse URLs, extract different components (e.g., protocol, hostname, pathname), and resolve relative URLs.
77. **What is the purpose of the querystring module in Node.js?**
    The querystring module in Node.js provides methods for working with query strings. It

allows you to parse query strings, stringify JavaScript objects into query string format, and perform query string-related operations.

78. **What is the purpose of the zlib module in Node.js?**
    The zlib module in Node.js provides compression and decompression functionalities using zlib, a software library for data compression. It allows you to compress data using various compression algorithms like deflate and gzip, as well as decompress compressed data.
79. **What is the purpose of the crypto module in Node.js?**
    The crypto module in Node.js provides cryptographic functionalities. It allows you to perform cryptographic operations like hashing, encrypting, decrypting, signing, and verifying data. It supports various cryptographic algorithms and provides a secure way to handle sensitive information.
80. **What is the purpose of the process.argv property in Node.js?**
    The process.argv property in Node.js is an array that contains the command-line arguments passed to the Node.js process. It allows you to access and process the arguments provided when running a Node.js script from the command line.
81. **How can you handle file operations in Node.js?**
    In Node.js, you can handle file operations using the fs module. It provides methods for interacting with the file system, such as reading and writing files, creating directories, deleting files, and more. You can use synchronous or asynchronous versions of these methods based on your requirements.
82. **What is the purpose of the path module in Node.js?**
    The path module in Node.js provides utilities for working with file paths. It allows you to manipulate file and directory paths, normalize paths, join paths, extract path components, and perform path-related operations in a cross-platform manner.
83. **What is the purpose of the util.promisify method in Node.js?**
    The util.promisify method in Node.js is used to convert callback-based functions into Promises. It takes a function that follows the Node.js callback style (taking a callback as the last argument) and returns a new function that returns a Promise instead. This simplifies working with asynchronous functions and enables the use of async/await syntax.
84. **What is the purpose of the Buffer class in Node.js?**
    The Buffer class in Node.js is used to handle binary data. It provides methods for creating, manipulating, and working with binary data, such as converting between different character encodings, performing bitwise operations, and working with streams of binary data.
85. **What is the purpose of the os module in Node.js?**
    The os module in Node.js provides utilities for interacting with the operating system. It allows you to access information about the system's CPUs, memory, network interfaces, system uptime, and more. It also provides platform-specific functionality and methods.
86. **What is the purpose of the cluster module in Node.js?**
    The cluster module in Node.js allows you to create a cluster of Node.js processes. It helps in utilizing multiple CPU cores by creating a child process for each core. It

provides an easy way to scale Node.js applications and improve performance by distributing the workload across multiple processes.

87. **What is the purpose of the http module in Node.js?**
    The http module in Node.js provides functionality for creating HTTP servers and making HTTP requests. It allows you to handle incoming HTTP requests, send HTTP responses, and interact with HTTP servers and clients.
88. **What is the purpose of the https module in Node.js?**
    The https module in Node.js is an extension of the http module and provides functionality for creating HTTPS servers and making secure HTTPS requests. It allows you to handle secure HTTP connections using SSL/TLS certificates.
89. **What is the purpose of the child_process module in Node.js?**
    The child_process module in Node.js allows you to create and control child processes. It provides methods for spawning new processes, communicating with them through standard input/output, and handling their execution. It helps in running external commands or scripts from within a Node.js application.
90. **What is the purpose of the stream module in Node.js?**
    The stream module in Node.js provides an interface for handling streaming data. It allows you to read from and write to streams, perform transformations on data streams, and pipe data between different streams. Streams provide an efficient way to process large amounts of data in chunks without loading everything into memory.
91. **What is the purpose of the crypto module in Node.js?**
    The crypto module in Node.js provides cryptographic functionality. It allows you to perform cryptographic operations such as hashing, encryption, decryption, signing, and verification. It supports various cryptographic algorithms and provides a secure way to handle sensitive information.
92. **What is the purpose of the assert module in Node.js?**
    The assert module inNode.js provides functions for assertion testing. It allows you to write assertions to check the values, types, and behavior of your code during development and testing. It helps ensure that your code behaves as expected and can catch potential bugs or issues early on.
93. **What is the purpose of the url module in Node.js?**
    The url module in Node.js provides utilities for working with URLs. It allows you to parse, format, and manipulate URLs, extract different components like the protocol, hostname, pathname, query parameters, and more. It helps in handling and manipulating URL strings efficiently.
94. **What is the purpose of the querystring module in Node.js?**
    The querystring module in Node.js provides utilities for working with query strings. It allows you to parse and stringify query strings, extract parameters, encode and decode URL components, and perform query string-related operations. It is commonly used in handling URL query parameters.
95. **What is the purpose of the zlib module in Node.js?**
    The zlib module in Node.js provides compression and decompression functionalities using the zlib library. It allows you to compress data using various compression

algorithms like deflate and gzip, as well as decompress compressed data. It helps in reducing the size of data for efficient storage and transmission.

96. **What is the purpose of the dns module in Node.js?**
    The dns module in Node.js provides methods for DNS (Domain Name System) operations. It allows you to perform DNS lookups, resolve hostnames to IP addresses, perform reverse DNS lookups, and more. It helps in resolving and working with domain names and IP addresses.
97. **What is the purpose of the util module in Node.js?**
    The util module in Node.js provides various utility functions and objects. It includes functions for working with objects, arrays, error handling, and more. It provides common utility functions that can be used in different parts of your application.
98. **What is the purpose of the readline module in Node.js?**
    The readline module in Node.js provides an interface for reading input from a readable stream, such as the command line. It allows you to prompt the user for input, read input line by line, and handle user interactions in a command-line application.
99. **What is the purpose of the process object in Node.js?**
    The process object in Node.js provides information and control over the current Node.js process. It allows you to access command-line arguments, environment variables, standard input/output streams, exit the process, listen for process events, and more. It provides a way to interact with the underlying operating system and manage the Node.js process.
100. **What is the purpose of the cluster module in Node.js?**
    The cluster module in Node.js allows you to create a cluster of Node.js processes. It helps in utilizing multiple CPU cores by creating a child process for each core. It provides an easy way to scale Node.js applications and improve performance by distributing the workload across multiple processes.
101. **What is the purpose of the os module in Node.js?**
    The os module in Node.js provides utilities for interacting with the operating system. It allows you to access information about the system's CPUs, memory, network interfaces, system uptime, and more. It provides platform-specific functionality and methods.
102. **What is the purpose of the http module in Node.js?**
    The http module in Node.js provides functionality for creating HTTP servers and making HTTP requests. It allows you to handle incoming HTTP requests, send HTTP responses, and interact with HTTP servers and clients. It forms the basis ofbuilding web applications and APIs in Node.js.
103. **What is the purpose of the https module in Node.js?**
    The https module in Node.js is an extension of the http module and provides functionality for creating HTTPS servers and making secure HTTPS requests. It allows you to handle secure HTTP connections using SSL/TLS certificates. It is used when you need to secure the communication between the server and the client.
104. **What is the purpose of the fs module in Node.js?**
    The fs module in Node.js provides functionality for interacting with the file system. It

allows you to read and write files, create directories, delete files, rename files, and perform other file-related operations. It provides synchronous and asynchronous methods for working with files and directories.

105. **What is the purpose of the stream module in Node.js?**

The stream module in Node.js provides an interface for handling streaming data. It allows you to read from and write to streams, perform transformations on data streams, and pipe data between different streams. Streams are used for processing large amounts of data in chunks, making it memory-efficient and suitable for handling data from sources like files, network connections, or HTTP requests.

106. **What is the purpose of the crypto module in Node.js?**

The crypto module in Node.js provides cryptographic functionality. It allows you to perform cryptographic operations such as hashing, encryption, decryption, signing, and verification. It supports various cryptographic algorithms and provides a secure way to handle sensitive information like passwords, authentication tokens, or data encryption.

107. **What is the purpose of the events module in Node.js?**

The events module in Node.js provides an implementation of the publisher-subscriber pattern. It allows objects to emit named events and register listeners that respond to those events. It facilitates efficient communication and decoupling between different parts of the application. The EventEmitter class is the core of the events module.

108. **What is the purpose of the util module in Node.js?**

The util module in Node.js provides various utility functions and objects. It includes functions for working with objects, arrays, error handling, and more. It provides common utility functions that can be used in different parts of your application. The util module also provides some useful utility classes and methods like promisify and inherits.

109. **What is the purpose of the path module in Node.js?**

The path module in Node.js provides utilities for working with file and directory paths. It allows you to manipulate file paths, join paths, resolve paths, extract components of a path, and perform path-related operations. The path module helps in working with file paths in a platform-independent manner.

110. **What is the purpose of the querystring module in Node.js?**

The querystring module in Node.js provides utilities for working with query strings. It allows you to parse query strings, stringify JavaScript objects into query string format, and perform query string-related operations like encoding and decoding URL components. It is commonly used for handling URL query parameters.

111. **What is the purpose of the dns module in Node.js?**

The dns module in Node.js provides methods for DNS (Domain Name System) operations. It allows you to perform DNS lookups, resolve hostnames to IP addresses, perform reverse DNS lookups, and more. It helps in working with domain names, IP addresses, and DNS-related operations.

112. **What is the purpose of the buffer module in Node.js?**

The buffer module in Node.js provides functionality for handling binary data. It allows you to create and manipulatebinary data buffers, perform operations like slicing,

copying, and concatenating buffers, and convert buffers to different encodings. It is commonly used when working with binary data, such as reading or writing files, network communications, or cryptographic operations.

113. **What is the purpose of the assert module in Node.js?**

The assert module in Node.js provides functions for assertion testing. It allows you to write assertions to check the values, types, and behavior of your code during development and testing. It helps ensure that your code behaves as expected and can catch potential bugs or issues early on.

114. **What is the purpose of the readline module in Node.js?**

The readline module in Node.js provides an interface for reading input from a readable stream, such as the command line. It allows you to prompt the user for input, read input line by line, and handle user interactions in a command-line application. It simplifies reading and processing user input in a structured manner.

115. **What is the purpose of the process object in Node.js?**

The process object in Node.js provides information and control over the current Node.js process. It allows you to access command-line arguments, environment variables, standard input/output streams, exit the process, listen for process events, and more. It provides a way to interact with the underlying operating system and manage the Node.js process.

116. **What is the purpose of the cluster module in Node.js?**

The cluster module in Node.js allows you to create a cluster of Node.js processes. It helps in utilizing multiple CPU cores by creating a child process for each core. It provides an easy way to scale Node.js applications and improve performance by distributing the workload across multiple processes.

117. **What is the purpose of the os module in Node.js?**

The os module in Node.js provides utilities for interacting with the operating system. It allows you to access information about the system's CPUs, memory, network interfaces, system uptime, and more. It provides platform-specific functionality and methods.

118. **What is the purpose of the http module in Node.js?**

The http module in Node.js provides functionality for creating HTTP servers and making HTTP requests. It allows you to handle incoming HTTP requests, send HTTP responses, and interact with HTTP servers and clients. It forms the basis of building web applications and APIs in Node.js.

119. **What is the purpose of the https module in Node.js?**

The https module in Node.js is an extension of the http module and provides functionality for creating HTTPS servers and making secure HTTPS requests. It allows you to handle secure HTTP connections using SSL/TLS certificates. It is used when you need to secure the communication between the server and the client.

120. **What is the purpose of the fs module in Node.js?**

The fs module in Node.js provides functionality for interacting with the file system. It allows you to read and write files, create directories, delete files, rename files, and perform other file-related operations. It provides synchronous and asynchronous methods for working with files and directories.

121. **What is the purpose of the stream module in Node.js?**
The stream module in Node.js provides an interface for handling streaming data. It allows you to read from and write to streams, perform transformations on data streams, and pipe data between different streams. Streams are used for processing large amounts of data in chunks, making it memory-efficient and suitable for handling data from sources like files, network connections, or HTTP requests.

122. **What is the purpose of the crypto module in Node.js?**
The crypto modulein Node.js provides cryptographic functionality. It allows you to perform cryptographic operations such as hashing, encryption, decryption, signing, and verification. It supports various cryptographic algorithms and provides a secure way to handle sensitive information like passwords, authentication tokens, or data encryption.

123. **What is the purpose of the events module in Node.js?**
The events module in Node.js provides an implementation of the publisher-subscriber pattern. It allows objects to emit named events and register listeners that respond to those events. It facilitates efficient communication and decoupling between different parts of the application. The EventEmitter class is the core of the events module.

124. **What is middleware in Express.js?**
Middleware in Express.js are functions that have access to the request and response objects. They can perform tasks like modifying request/response objects, executing additional code, or passing control to the next middleware function. Middleware functions are executed in a sequential order as defined.

125. **What is the purpose of the body-parser middleware in Express.js?**
The body-parser middleware in Express.js is used to parse the body of incoming requests. It extracts data from the request body and makes it available in req.body for further processing.

126. **How do you handle routing in Express.js?**
In Express.js, you can define routes using the app.get(), app.post(), app.put(), app.delete(), and other methods provided by the Express application. Each route maps to a specific URL and specifies a callback function that handles the request and generates the response.

127. **Explain the concept of template engines in Express.js.**
Template engines in Express.js are used to dynamically generate HTML pages by combining data with predefined HTML templates. Express.js supports various template engines like EJS, Pug (formerly Jade), Handlebars, etc. These engines allow you to embed dynamic data within the HTML template, making it easier to generate dynamic web pages.

128. **What is the difference between app.use() and app.METHOD() in Express.js?**

129. app.use() is a middleware function that is executed for every incoming request, regardless of the HTTP method. It is commonly used for tasks like setting up middleware, defining routes, and handling error middleware.

130. app.METHOD() (e.g., app.get(), app.post(), app.put()) is used to define route-specific middleware that is executed only for the specified HTTP method and route.

131. **What is the purpose of the dotenv module in Node.js?**
The dotenv module in Node.js allows you to load environment variables from a .env

file into the process.env object. It provides an easy way to manage and access configuration settings for your Node.js application.

132. **What is the purpose of the async module in Node.js?**
The async module in Node.js provides a powerful set of utility functions for handling asynchronous operations. It allows you to control the flow of asynchronous code using functions like async.waterfall(), async.parallel(), async.series(), etc.

133. **What is the purpose of the mongoose module in Node.js?**
The mongoose module is an Object-Data Modeling (ODM) library for MongoDB and Node.js. It provides an elegant and intuitive way to interact with MongoDB databases, defining schemas, models, and performing database operations.

134. **Explain the concept of Promises in Node.js.**
Promises in Node.js provide a cleaner way to handle asynchronous operations. They represent the eventual completion (or failure) of an asynchronous operation and allow you to chain multiple asynchronous operations together, making code more readable and avoiding callback hell.

135. **What are the different states of a Promise in Node.js?**
A Promise in Node.js can be in one of three states:

136. Pending: The initial state of a Promise. It indicates that the asynchronous operation is still ongoing and has not yet been fulfilled or rejected.

137. Fulfilled: The state of a Promise when the asynchronous operation has completed successfully, and the associated value (result) is available.

138. Rejected: The state of a Promise when the asynchronous operation has encountered an error or failure, and the associated reason (error) is available.

139. **How do you handle errors in Promises?
In Promises, you can handle errors using the catch method or by chaining the catch method after the then method. This allows you to handle any errors that occur during the asynchronous operation and perform appropriate error handling or fallback actions.

140. **What is the purpose of the async/await feature in Node.js?**
The async/await feature in Node.js provides a more synchronous-style way to write asynchronous code. It allows you to write asynchronous operations using synchronous-like syntax, making it easier to read, write, and maintain asynchronous code.

141. **What is the purpose of the EventEmitter class in Node.js?**
The EventEmitter class in Node.js provides an implementation of the publisher-subscriber pattern. It allows objects to emit named events and register listeners that respond to those events, enabling efficient communication and decoupling between different parts of the application.

142. **What is the purpose of the process object in Node.js?**
The process object in Node.js provides information and control over the current Node.js process. It allows you to access command-line arguments, environment variables, exit the process, listen for process events, and more.

143. **How do you handle file uploads in Node.js?**
To handle file uploads in Node.js, you can use middleware like multer. Multer is a popular middleware that handles multipart/form-data, allowing you to process file uploads in your Express.js application.

144. **What is the purpose of the os module in Node.js?**
The os module in Node.js provides a set of utility methods for interacting with the

operating system. It allows you to access information about the system's CPUs, memory, network interfaces, and more.

145. **How do you implement caching in Node.js?**
Caching in Node.js can be implemented using various techniques, such as in-memory caching, using caching libraries like Redis or Memcached, or utilizing HTTP caching headers like Cache-Control and ETag. Caching helps improve performance by storing frequently accessed data and reducing the load on the server.

146. **What is the purpose of the child_process module in Node.js?**
The child_process module in Node.js allows you to create and control child processes. It provides methods to spawn new processes, communicate with them through standard input/output, and handle their execution.

147. **What is the purpose of the util module in Node.js?**
The util module in Node.js provides various utility functions and objects that are useful for debugging, error handling, and other common tasks. It includes functions like promisify for converting callback-based functions into Promises, inherits for prototypal inheritance, and more.

148. **What are the differences between setTimeout and setImmediate in Node.js?**

149. setTimeout schedules a function to be executed after a specified delay (in milliseconds). It adds the callback to the timers phase of the event loop.

150. setImmediate schedules a function to be executed in the next iteration of the event loop, immediately after the I/O callbacks phase.

151. **What is the purpose of the URL module in Node.js?**
The URL module in Node.js provides utilities for working with URLs. It allows you to parse, manipulate, and construct URLs, extract different components (e.g., hostname, path, query parameters), and perform URL-related operations.

152. **What is the purpose of the cluster module in Node.js?**
The cluster module in Node.js allows you to create a cluster of Node.js processes to take advantage of multiple CPU cores. It enables load balancingand improves the scalability and performance of Node.js applications by distributing the workload among multiple worker processes.

153. **What is the purpose of the http module in Node.js?**
The http module in Node.js provides functionality for creating HTTP servers and making HTTP requests. It allows you to handle incoming HTTP requests, send HTTP responses, and interact with HTTP servers and clients.

154. **What is the purpose of the https module in Node.js?**
The https module in Node.js extends the http module and provides functionality for creating HTTPS servers and making secure HTTPS requests. It allows you to handle secure HTTP connections using SSL/TLS certificates.

155. **How can you handle concurrent requests in Node.js?**
In Node.js, concurrent requests can be handled efficiently due to its asynchronous and non-blocking nature. By using event-driven programming, callbacks, promises, or async/await, you can handle multiple requests concurrently without blocking the execution of other requests.

156. **What is the purpose of the net module in Node.js?**
The net module in Node.js provides functionality for creating TCP servers and clients. It allows you to create TCP connections, send and receive data over TCP sockets, and build TCP-based networking applications.

157. **What is the purpose of the dns module in Node.js?**
The dns module in Node.js provides methods for DNS (Domain Name System) operations. It allows you to perform DNS lookups, resolve hostnames to IP addresses, perform reverse DNS lookups, and more.
158. **What is the purpose of the url module in Node.js?**
The url module in Node.js provides methods for URL resolution and parsing. It allows you to parse URLs, extract different components (e.g., protocol, hostname, pathname), and resolve relative URLs.
159. **What is the purpose of the querystring module in Node.js?**
The querystring module in Node.js provides methods for working with query strings. It allows you to parse query strings, stringify JavaScript objects into query string format, and perform query string-related operations.
160. **What is the purpose of the zlib module in Node.js?**
The zlib module in Node.js provides compression and decompression functionalities using zlib, a software library for data compression. It allows you to compress data using various compression algorithms like deflate and gzip, as well as decompress compressed data.
161. **What is the purpose of the crypto module in Node.js?**
The crypto module in Node.js provides cryptographic functionalities. It allows you to perform cryptographic operations like hashing, encrypting, decrypting, signing, and verifying data. It supports various cryptographic algorithms and provides a secure way to handle sensitive information.
162. **What is the purpose of the process.argv property in Node.js?**
The process.argv property in Node.js is an array that contains the command-line arguments passed to the Node.js process. It allows you to access and process the arguments provided when running a Node.js script from the command line.
163. **How can you handle file operations in Node.js?**
In Node.js, you can handle file operations using the fs module. It provides methods for interacting with the file system, such as reading and writing files, creating directories, deleting files, and more. You can use synchronous or asynchronous versions of these methods based on your requirements.
164. **What is the purpose of the path module in Node.js?**
The path module in Node.js provides utilities for working with file and directory paths. It allows you to manipulate file and directory paths, normalize paths, join paths, extract path components, and perform path-related operations ina cross-platform manner.
165. **What is the purpose of the util.promisify method in Node.js?**
The util.promisify method in Node.js is used to convert callback-based functions into Promises. It takes a function that follows the Node.js callback style (taking a callback as the last argument) and returns a new function that returns a Promise instead. This simplifies working with asynchronous functions and enables the use of async/await syntax.
166. **What is the purpose of the Buffer class in Node.js?**
The Buffer class in Node.js is used to handle binary data. It provides methods for creating, manipulating, and working with binary data, such as converting between different character encodings, performing bitwise operations, and working with streams of binary data.
167. **What is the purpose of the os module in Node.js?**
The os module in Node.js provides utilities for interacting with the operating system. It

allows you to access information about the system's CPUs, memory, network interfaces, system uptime, and more. It also provides platform-specific functionality and methods.

168. **What is the purpose of the cluster module in Node.js?**
The cluster module in Node.js allows you to create a cluster of Node.js processes. It helps in utilizing multiple CPU cores by creating a child process for each core. It provides an easy way to scale Node.js applications and improve performance by distributing the workload across multiple processes.

169. **What is the purpose of the http module in Node.js?**
The http module in Node.js provides functionality for creating HTTP servers and making HTTP requests. It allows you to handle incoming HTTP requests, send HTTP responses, and interact with HTTP servers and clients.

170. **What is the purpose of the https module in Node.js?**
The https module in Node.js is an extension of the http module and provides functionality for creating HTTPS servers and making secure HTTPS requests. It allows you to handle secure HTTP connections using SSL/TLS certificates.

171 **What is the purpose of the child_process module in Node.js?**
The child_process module in Node.js allows you to create and control child processes. It provides methods for spawning new processes, communicating with them through standard input/output, and handling their execution. It helps in running external commands or scripts from within a Node.js application.

172. **What is the purpose of the stream module in Node.js?**
The stream module in Node.js provides an interface for handling streaming data. It allows you to read from and write to streams, perform transformations on data streams, and pipe data between different streams. Streams provide an efficient way to process large amounts of data in chunks without loading everything into memory.

173. **What is the purpose of the crypto module in Node.js?**
The crypto module in Node.js provides cryptographic functionalities. It allows you to perform cryptographic operations like hashing, encrypting, decrypting, signing, and verifying data. It supports various cryptographic algorithms and provides a secure way to handle sensitive information.

174. **What is the purpose of the assert module in Node.js?**
The assert module in Node.js provides functions for assertion testing. It allows you to write assertions to check the values, types, and behavior of your code during development and testing. It helps ensure that your code behaves as expected and can catch potential bugs or issues early on.

175. **What is the purpose of the readline module in Node.js?**
The readline module in Node.js provides an interface for reading input from a readable stream, such as the command line. It allows you to prompt the userfor input, read input line by line, and handle user interactions in a command-line application.

176. **What is the purpose of the process object in Node.js?**
The process object in Node.js provides information and control over the current Node.js process. It allows you to access command-line arguments, environment variables, standard input/output streams, exit the process, listen for process events, and more. It provides a way to interact with the underlying operating system and manage the Node.js process.

177. **What is the purpose of the util module in Node.js?**
The util module in Node.js provides various utility functions and objects. It includes

functions for working with objects, arrays, error handling, and more. It provides common utility functions that can be used in different parts of your application.

178. **What is the purpose of the path module in Node.js?**
The path module in Node.js provides utilities for working with file and directory paths. It allows you to manipulate file and directory paths, join paths, resolve paths, extract components of a path, and perform path-related operations. The path module helps in working with file paths in a platform-independent manner.

179. **What is the purpose of the querystring module in Node.js?**
The querystring module in Node.js provides utilities for working with query strings. It allows you to parse query strings, stringify JavaScript objects into query string format, and perform query string-related operations. It is commonly used for handling URL query parameters.

180. **What is the purpose of the zlib module in Node.js?**
The zlib module in Node.js provides compression and decompression functionalities using zlib, a software library for data compression. It allows you to compress data using various compression algorithms like deflate and gzip, as well as decompress compressed data.