These questions test understanding of event handling, promise-based asynchronous functions, and error handling in Node.js.

```
const getRequestBody = function (req: IncomingMessage): Promise<string> {
  return new Promise((resolve, reject) => {
    let body = '';
    req.on('data', (chunk) => {
      body += chunk.toString();
    });
    req.on('end', () => {
      resolve(body);
    });
    req.on('error', (err) => {
      reject(err);
    });
  });
};
```

1. **What is the primary purpose of the getRequestBody function?**
2. **Which of the following best describes the type of the getRequestBody function?**
3. **In the code, what event is used to collect chunks of data from the incoming request?**
4. **What does body += chunk.toString(); do in the context of this function?**
5. **When is the resolve function called in this code?**
6. **Which event triggers the reject function in the code?**
7. **What is the type of the req parameter in the getRequestBody function?**
8. **If there is an error in reading the request body, what will happen?**
9. **Which of the following correctly describes the flow of this function?**

10. **What will happen if there are no errors and no data chunks are received in the data event?**

Here’s a set of 30 multiple-choice questions based on the provided code and its functionalities, focusing on key concepts in Node.js, HTTP request handling, TypeScript, and in-memory storage.

---

11. **What is the purpose of the IncomingMessage and ServerResponse imports in the code?**
12. **Which module is used to create the server in this code?**
13. **What type is assigned to the hostname constant?**
14. **What value is assigned to hostname?**
15. **Which endpoint returns all tasks in the current task list?**
16. **What HTTP method is used to create a new task?**
17. **How does the server identify a request targeting a single task by ID?**
18. **Which HTTP status code is returned when a task is successfully created?**
19. **What does the getRequestBody function do?**
20. **What is the purpose of the taskIdCounter variable?**

21. **What type of storage is used for the tasks in this code?**
22. **Which HTTP method is used to retrieve a single task by ID?**
23. **What response status code is returned when a task with a specified ID is not found?**
24. **What is checked in the isTaskUrl function?**
25. **How does the code handle errors in parsing the request body?**
26. **Which property is NOT included in the Task interface?**
27. **What does the server return if an endpoint does not match any defined routes?**
28. **What happens when the PUT method is used with a valid task ID?**
29. **set to false if not provided when creating a new task?**
30. **What HTTP method is used to delete a task by ID?**
31. **What status code does the server return if an invalid task ID is provided?**
32. **In the getRequestBody function, which event is used to read the data in chunks?**
33. **What does the server return when a task is successfully deleted?**
34. **If a request body is invalid in a POST request, what status code is returned?**
35. **Where is the task ID extracted in the code?**
36. **What is the purpose of res.end() in this code?**
37. **What is the type of the tasks array?**
38. **Which header is set when returning JSON data?**

39. **How does the server respond if a request is sent to an undefined endpoint?**
40. **Which of the following is used to parse the request body in this code?**
41. **What is the purpose of setting up a separate tsconfig.test.json file?**
42. **Which Jest configuration option specifies the directory where Jest will look for test files?**
43. **Which command is typically used to run Jest tests in a Node.js project?**
44. **What does the Jest testEnvironment configuration option control?\**
45. **In Jest, what is the purpose of using beforeEach in a test suite?**
46. **Which TypeScript compiler option should be set in tsconfig.json to avoid checking .test.ts files?**
47. **In the test case for POST /tasks, what should the test expect when the title field is missing?**
48. **Which validation function can be used to ensure a string is within a specific length?**
49. **What is the purpose of the errors array in the validation response format { success, payload, msg, errors }?**
50. **If completed is a boolean field in the POST /tasks API, what should happen if a non-boolean value is sent?**
51. **Which of the following configurations is typically used to tell Jest to transform TypeScript files?**
52. **In Jest, what is the purpose of expect()?**
53. **In Jest, which assertion checks if two values are deeply equal?**
54. **How do you ignore test files in TypeScript using tsconfig.json?**
55. **When using Jest with TypeScript, what role does ts-jest play?**
56. **What Jest function would you use to verify that a specific function was called during a test?**
57. **In Jest, what is a "mock"?**
58. **If Jest is not recognizing .test.ts files, which configuration would you check first?**
59. **Which configuration setting in Jest tells it to use tsconfig.test.json for custom TypeScript settings during testing?**
60. **What is the main purpose of validation in API development?**
61. **Which HTTP status code is commonly used when validation fails on an API request?**
62. **In a validation schema, which method checks if a field is required?**
63. **What would be the expected HTTP status code and response message if a mandatory field, like "title," is missing in a POST request?**
64. **Which library is often used in Node.js to simplify validation rules for fields?**
65. **In testing validations, what assertion is used to check that a field exists within the response payload?**
66. **What would be an appropriate test for validating a maximum length requirement for the description field in the POST /tasks endpoint?**

67. **In the response { success: false, payload: {}, msg: 'Validation Error', errors: ['title is required'] }, what does errors represent?**
68. **When using supertest to test a validation error, which of the following checks is relevant?**
69. **How do you ensure that the msg field in the response indicates a successful task creation in the POST /tasks test case?**
70. **In the PUT /tasks/:id endpoint test, which validation check should be applied if id is invalid?**
71. **For DELETE /tasks/:id, what should the test expect if the task ID is not found?**
72. **What is the purpose of using toMatchObject() when checking the response body in tests?**
73. **Which of the following is a valid way to test for a success: false response on validation failure?**
74. **If the API returns errors: ['Invalid data format'] on a PUT request, what should the test assert?**
75. **Why is it important to include validation tests for each required field in API testing?**