1. True/False: Flutter provides built-in support for API integration.

   - Answer: False. Flutter does not have built-in support for API integration, but it offers packages and libraries to help with it.

2. True/False: You can make API requests in Flutter using the `http` package.

   - Answer: True. The `http` package is commonly used to make HTTP requests in Flutter.

3. True/False: Flutter can only communicate with RESTful APIs.

   - Answer: False. Flutter can communicate with RESTful APIs, GraphQL APIs, and other web services.

4. True/False: API integration in Flutter requires the use of external libraries and packages.

   - Answer: True. You typically need to use external libraries like `http` or `dio` for API integration in Flutter.

5. True/False: Flutter widgets like `FutureBuilder` are useful for handling asynchronous API calls.

   - Answer: True. Widgets like `FutureBuilder` are helpful for managing asynchronous API calls and updating the UI accordingly.

6. True/False: Flutter supports both GET and POST requests for API integration.

   - Answer: True. Flutter supports both GET and POST requests, along with other HTTP methods.

7. True/False: Error handling is not necessary when making API requests in Flutter.

   - Answer: False. Proper error handling is essential when making API requests to handle network issues or server errors.

8. True/False: You can use the `async` and `await` keywords in Flutter to work with asynchronous API calls.

   - Answer: True. You can use `async` and `await` to work with asynchronous API calls in Flutter for cleaner code.

9. True/False: Flutter's `Future` class is commonly used to represent asynchronous operations when dealing with APIs.

- Answer: True. The `Future` class is often used to represent asynchronous operations, including API requests.

10. True/False: You must always use third-party state management libraries like Provider or Riverpod for API integration in Flutter.

- Answer: False. While they can be helpful, they are not mandatory for API integration; Flutter's built-in `setState` can also be used.

11. True/False: You can cache API responses in Flutter to improve performance.

- Answer: True. Caching API responses can reduce the number of network requests and improve app performance.

12. True/False: Flutter provides built-in tools for automatic API response parsing.

- Answer: False. You need to manually parse API responses using libraries like `dart:convert` or `json_serializable`.

13. True/False: Flutter allows you to mock API responses for testing purposes.

- Answer: True. You can use packages like `http` to mock API responses during testing.

14. True/False: Flutter's `http` package automatically handles pagination for API endpoints.

- Answer: False. Pagination usually needs to be implemented manually based on the API's pagination structure.

15. True/False: You should always store API keys directly in your Flutter source code.

- Answer: False. Storing API keys directly in source code is not recommended for security reasons; use environment variables or a secure storage solution.

16. True/False: Flutter's `dio` package is a more feature-rich alternative to `http` for API integration.

- Answer: True. `dio` offers more features like request cancellation, interceptors, and FormData handling compared to `http`.

17. True/False: Flutter's `SharedPreference` class can be used to store API tokens securely.

- Answer: False. `SharedPreference` is not a secure way to store API tokens. Consider using more secure solutions like `flutter_secure_storage`.

18. True/False: Flutter widgets can automatically update when API data changes.

- Answer: False. Widgets do not automatically update when API data changes; you need to manage state and trigger updates manually.

19. True/False: Flutter has a built-in mechanism for rate-limiting API requests.

- Answer: False. Rate limiting for API requests must be implemented manually based on the API's rate-limiting policies.

20. True/False: Flutter can handle real-time data updates from APIs using WebSocket connections.

- Answer: True. Flutter can handle real-time data updates from APIs by using WebSocket connections or packages like `web_socket_channel`.

Question 2

1. What does API stand for in the context of Flutter?

a) Application Programming Interface

b) Android Programming Interface

c) Application Protocol Interface

d) Application Program Interface

Answer: a) Application Programming Interface

2. Which Flutter package is commonly used for making HTTP requests to APIs?

a) http

b) api_requester

c) network_manager

d) data_connector

Answer: a) http

3. In Flutter, which method is commonly used to perform asynchronous API requests?

a) Future.delayed()

b) asyncRequest()

c) FutureBuilder()

d) await

Answer: d) await

4. Which HTTP method is typically used for retrieving data from an API?

a) POST

b) PUT

c) GET

d) DELETE

Answer: c) GET

5. What is the primary purpose of the `FutureBuilder` widget in Flutter when dealing with API integration?

a) To define API endpoints

b) To parse JSON responses

c) To handle asynchronous tasks and update the UI

d) To perform HTTP requests

Answer: c) To handle asynchronous tasks and update the UI

6. Which HTTP status code indicates that a resource was successfully created on the server?

a) 200 OK

b) 201 Created

c) 204 No Content

d) 400 Bad Request

Answer: b) 201 Created

7. When parsing JSON data from an API response, what is a common package used to convert JSON strings into Dart objects?

a) json_serializable

b) json_parser

c) dart_json

d) json_converter

Answer: a) json_serializable

8. What is the purpose of an API key when making requests to some APIs?

a) To specify the HTTP method

b) To authenticate and authorize access to the API

c) To define API endpoints

d) To store API response data

Answer: b) To authenticate and authorize access to the API

9. Which widget is used to display a loading indicator while waiting for data from an API request?

a) CircularProgressIndicator

b) ProgressIndicator

c) LoadingWidget

d) LoaderIndicator

Answer: a) CircularProgressIndicator

10. In Flutter, what is the purpose of the `Future` class when working with APIs?

a) To create a new HTTP request

b) To handle asynchronous operations and represent a potential value or error

c) To parse JSON data

d) To define API endpoints

Answer: b) To handle asynchronous operations and represent a potential value or error

11. Which HTTP status code indicates that the requested resource could not be found on the server?

a) 200 OK

b) 401 Unauthorized

c) 404 Not Found

d) 500 Internal Server Error

Answer: c) 404 Not Found

12. To make an authenticated API request, which header is commonly used to send the authentication token?

a) Authorization

b) Authentication-Token

c) Auth-Token

d) API-Token

Answer: a) Authorization

13. When should you use the `http` package in Flutter for API requests instead of the `dio` package?

a) When you need advanced features like caching and interceptors

b) When you need to perform complex data transformations

c) When you only need to make simple HTTP requests

d) When you need WebSocket support

Answer: c) When you only need to make simple HTTP requests

14. Which method is used to handle errors when making API requests using the `http` package in Flutter?

a) onError()

b) catchError()

c) onFailure()

d) handleErrors()

Answer: b) catchError()

15. What is the purpose of a RESTful API?

a) To perform remote procedure calls

b) To create user interfaces

c) To represent resources as URLs and use HTTP methods for CRUD operations

d) To manage database connections

Answer: c) To represent resources as URLs and use HTTP methods for CRUD operations

16. Which HTTP method is used to update an existing resource on the server in a RESTful API?

a) POST

b) PUT

c) PATCH

d) DELETE

Answer: c) PATCH

17. What is the primary benefit of using asynchronous API requests in Flutter?

a) Improved security

b) Reduced code complexity

c) Faster network speed

d) Preventing UI freezes while waiting for responses

Answer: d) Preventing UI freezes while waiting for responses

18. In Flutter, how can you pass data from an API request to another screen or widget?

a) Using the Navigator class

b) By using the global variable

c) By embedding the data in the URL

d) By using the SharedPreferences package

Answer: a) Using the Navigator class

19. Which package is commonly used for managing and storing API response data in Flutter?

a) redux

b) provider

c) hive

d) sqflite

Answer: c) hive

20. What is the purpose of the `http.Response` object in Flutter when making API requests?

a) To define API endpoints

b) To represent the HTTP request

c) To store API request data

d) To represent the HTTP response and its properties

Answer: d) To represent the HTTP response and its properties