

Week 9 - LAQ's

Instructions

Explain the methods used in REST APIs to perform actions on resources:

REST (Representational State Transfer) APIs utilize standard HTTP methods to perform actions on resources. Each method corresponds to a specific action that can be taken on the resource identified by a URI (Uniform Resource Identifier). Here's a detailed explanation of the most commonly used HTTP methods in REST APIs:

1. GET

- Purpose: Retrieve data from the server.
- Usage: This method is used to request data from a specified resource. It should not modify any data on the server.
- Idempotent: Yes (multiple identical requests should have the same effect as a single request).
- Example:
- GET /users retrieves a list of users.
- GET /users/1 retrieves the user with ID 1.

2. POST

- Purpose: Create a new resource on the server.

- Usage: This method is used to send data to the server to create a new resource. The server assigns a unique identifier for the new resource.
- Idempotent: No (multiple identical requests may create multiple resources).
- Example:
- POST /users with a request body containing user details creates a new user.

3. PUT

- Purpose: Update an existing resource or create a new resource if it does not exist.
- Usage: This method is used to send data to the server to update a specified resource. If the resource does not exist, it may create it.
- Idempotent: Yes (multiple identical requests should have the same effect as a single request).
- Example:
- PUT /users/1 with a request body containing updated user details updates the user with ID 1.

4. PATCH

- Purpose: Partially update an existing resource.
- Usage: This method is used to apply partial modifications to a resource. Unlike PUT, which requires the entire resource representation, PATCH only requires the fields that need to be updated.
- Idempotent: Yes (multiple identical requests should have the same effect as a single request).

- Example:
- PATCH /users/1 with a request body containing { "age": 30 } updates only the age of the user with ID 1.

5. DELETE

- Purpose: Remove a resource from the server.
- Usage: This method is used to delete a specified resource from the server.
- Idempotent: Yes (multiple identical requests should have the same effect as a single request).
- Example:
- DELETE /users/1 removes the user with ID 1 from the server.

6. OPTIONS

- Purpose: Describe the communication options for the target resource.
- Usage: This method is used to determine the HTTP methods that are supported by the server for a specific resource. It is often used in CORS (Cross-Origin Resource Sharing) scenarios.
- Idempotent: Yes.
- Example:
- OPTIONS /users might return a response indicating that GET, POST, and DELETE methods are allowed.

Summary of Idempotency

- Idempotent Methods: GET, PUT, DELETE, PATCH (performing the same operation multiple times results in the same state).

- Non-Idempotent Method: POST (performing the same operation multiple times can result in different states).

Conclusion

These HTTP methods provide a standardized way to interact with resources in REST APIs. By adhering to these conventions, developers can create APIs that are intuitive and predictable, making it easier for clients to consume and interact with them effectively.