Rest API

MERN STACK : Content

- Introduction to Rest API's
- HTTP verbs
- Questionnaire

Introduction to Rest - API's

- A REST API (Representational State Transfer - Application Programming Interface) is a set of rules that developers follow when they create APIs.
- APIs allow two software applications to communicate with each other over the web.

Key Concepts of REST API:

- **Resources:** In REST, everything is treated as a resource, which could be any object, data, or service. Resources are identified by URLs (Uniform Resource Locators). For example, a user's profile or a blog post can be a resource.

Introduction to Rest - API's

- **HTTP Methods:** REST APIs use standard HTTP methods to perform actions on resources. These are:
  - GET: Retrieve data from a resource.
  - POST: Create a new resource.
  - PUT: Update an existing resource.
  - DELETE: Remove a resource.
- **Statelessness:** Each request from the client to the server must contain all the information the server needs to fulfill the request. The server does not store any state between requests, making each API call independent of others.

Introduction to Rest - API's

- **CRUD Operations:** REST APIs often follow the CRUD model (Create, Read, Update, Delete) to manage resources:
  - Create: Corresponds to POST
  - Read: Corresponds to GET
  - Update: Corresponds to PUT or PATCH
  - Delete: Corresponds to DELETE
- **JSON/XML Data Format:** REST APIs typically use JSON (JavaScript Object Notation) as the format for sending and receiving data because it's lightweight and easy to read. Some APIs might also support XML.

Introduction to Rest - API's

Example of a REST API Call: Let's say you have a REST API to manage blog posts. An example interaction could be:

- To retrieve all posts: GET /posts
- To retrieve a specific post: GET /posts/postid
- To create a new post: POST /posts with post details in the request body.
- To update a post: PUT /posts/postid with updated post details.
- To delete a post: DELETE /posts/postid

Introduction to Rest - API's

Benefits of REST APIs: Here are some of the benefits of REST APIs

- Scalability: Since REST APIs are stateless, they can handle large numbers of requests more easily, making them scalable.
- Flexibility: REST APIs can handle different types of calls, return different data formats, and even change structure with new versions.
- Interoperability: REST is language-agnostic, allowing it to be used by different systems and platforms.

HTTP verbs

- HTTP verbs (also known as HTTP methods) are fundamental building blocks of REST APIs.
- They define the type of action the client wants to perform on a resource, typically related to managing data (create, read, update, delete).
- Each verb corresponds to a different action.

HTTP verbs(Continued...)

Here's a breakdown of the most commonly used HTTP verbs:

- GET: Retrieve data from a server. This method is idempotent, meaning multiple identical requests will yield the same result.
- POST: Send data to the server to create a new resource. This method is not idempotent, as repeated requests can create multiple resources.
- PUT: Update an existing resource or create it if it does not exist. This method is idempotent.
- PATCH: Partially update an existing resource. Unlike PUT, which replaces the entire resource, PATCH modifies only the specified fields.
- DELETE: Remove a resource from the server. This method is also idempotent; deleting the same resource multiple times has the same effect as deleting it once.
- HEAD: Similar to GET but only retrieves the headers, not the body. Useful for checking resource availability without transferring data.

HTTP verbs(Continued...)

- OPTIONS: Describe the communication options for the target resource, allowing clients to understand what methods are supported.
- TRACE: Echo back the received request, used mainly for diagnostic purposes.
- CONNECT: Establish a tunnel to the server identified by the target resource, often used with HTTPS.

Questionnaire

Questions

Thank you