Exploring RESTful Services with JSONPlaceholder API

Objective:
Practice making HTTP GET, POST, PUT, and DELETE requests using Postman to understand how RESTful services work.

Preparation:
- Install Postman or any similar API testing tool.
- Familiarize yourself with JSONPlaceholder, a free service that provides fake online REST data: JSONPlaceholder

Part 1: Fetching Data with GET
- Open Postman.
- Create a new request tab.
- Set the HTTP method to GET.
- Enter the URL https://jsonplaceholder.typicode.com/posts to fetch all posts.
- Send the request and observe the response.
- Repeat the process for https://jsonplaceholder.typicode.com/posts/1 to fetch a single post.

Part 2: Creating Data with POST
- Change the HTTP method to POST.
- Enter the URL https://jsonplaceholder.typicode.com/posts.
- Go to the 'Body' tab, select 'raw', and choose 'JSON' from the dropdown.

Enter the following JSON data:

```
{
 "title"  "foo"
 "body"  "bar"
 "userId"  1
}
```

- Send the request and observe the response, noting the new ID assigned.

Part 3: Updating Data with PUT
- Change the HTTP method to PUT.
- Enter the URL https://jsonplaceholder.typicode.com/posts/1 to update the first post.
- In the 'Body' tab, input new JSON data for the title or body.
- Send the request and observe the response, noting how the data is updated.

Part 4: Deleting Data with DELETE

- Change the HTTP method to DELETE.
- Enter the URL https://jsonplaceholder.typicode.com/posts/1 to delete the first post.
- Send the request and observe the response, noting that it should return an empty JSON object.

Reflection Questions:

- What status code do you get for each type of request?
- What happens if you try to GET the data you just deleted?
- Why is it important to have different methods like GET, POST, PUT, and DELETE?

Challenge:

- Try to add a new comment to a post using POST to **https://jsonplaceholder.typicode.com/comments**.
- Update a comment using PUT.
- Delete a comment using DELETE.
- Think about how you would design your own API. What kind of data would it serve? What would the endpoints be?

Exploring the Giphy API with Postman

Objective: Learn to make various HTTP requests to the Giphy API and understand the responses.

Prerequisites:
- Install Postman or any similar API testing tool.
- Obtain an API key from Giphy by signing up at Giphy Developers and creating an app.

Step 1: Setup and Authentication
- Create a new request in Postman.
- Set the request type to GET.
- Use the base URL for Giphy's API: http://api.giphy.com/v1/gifs/
- Append your API key to all requests as a query parameter: ?api_key=YOUR_API_KEY

Step 2: Search for GIFs
Objective: Use the search endpoint to find GIFs related to a keyword.

- GET Request URL: http://api.giphy.com/v1/gifs/search?api_key=YOUR_API_KEY&q=funny+cats
- Send the request and explore the JSON response. Identify the structure, especially how GIFs are represented.

Step 3: Trending GIFs
Objective: Retrieve trending GIFs.

- GET Request URL: http://api.giphy.com/v1/gifs/trending?api_key=YOUR_API_KEY
- Analyze the response to understand what makes these GIFs "trending".

Step 4: Random GIF
Objective: Fetch a random GIF.

- GET Request URL: http://api.giphy.com/v1/gifs/random?api_key=YOUR_API_KEY
- Discuss the randomness - refresh the request multiple times to see different results.

Reflection and Discussion
- Discuss the limitations and possibilities of working with third-party APIs.
- Reflect on how different HTTP methods serve various purposes in web development.