API Session – Chapter 8

- API Chaining -

# API Chaining: Linking API Calls Together

**API chaining** is a technique where the output of one API call is used as input for another API call. This creates a sequence of interconnected API requests, often automating a multi-step process.

## How it works?

* **Initiate the first API call:** This call provides the initial data or parameters.
* **Process the response:** Extract relevant information from the response.
* **Use the extracted data:** Pass this data as input to the next API call.
* **Repeat:** Continue this process for multiple API calls as needed.

## Benefits of API Chaining

* **Automation:**

Reduces manual intervention and potential errors.

* **Efficiency:**

Streamlines complex workflows by combining multiple steps into a single process.

* **Data consistency:**

Ensures data integrity by passing information directly between APIs.

* **Complex scenario testing:**

Allows for testing of real-world use cases involving multiple API interactions.

## Common Use Cases:

* **E-commerce:** Calculating shipping costs, processing payments, and updating inventory.
* **Social media:** Creating a post, adding images, and getting user interactions.
* **Data processing:** Fetching data from one source, transforming it, and storing it in another system.

## Tools for API Chaining:

* **Postman:** Widely used for API testing and development, offers scripting capabilities for chaining.
* **Programming languages:** Languages like Python, JavaScript, and Java can be used to build custom chaining logic.

# Go Rest API

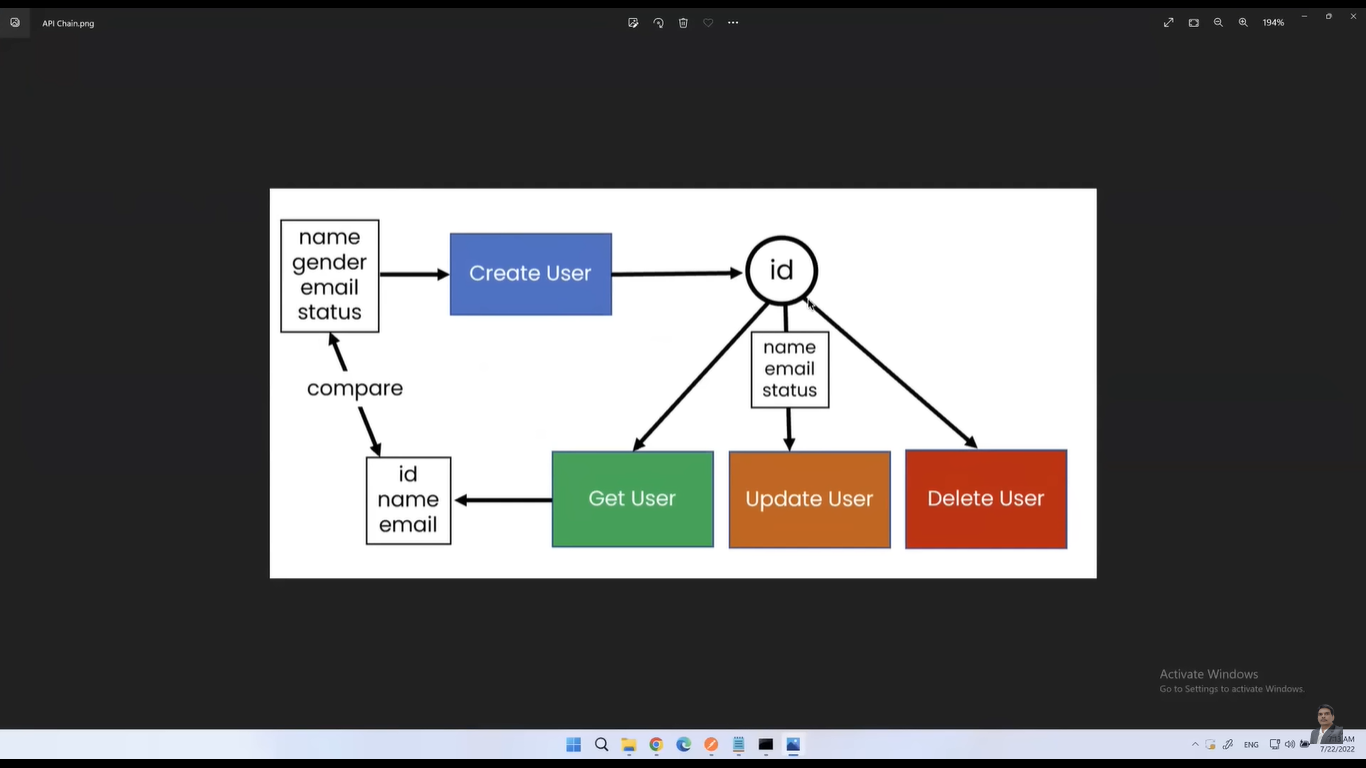
Website Link - <https://gorest.co.in/>

This is a sample website provides an online service called GO REST. It discusses how to use the service's API and its features. Users can make API calls to create, retrieve, update, and delete user records. The service also includes a GraphQL API. It discusses the endpoints for users, posts, comments, and todos.

* Users can be created, viewed, updated, and deleted.
* Posts, comments, and todos can be retrieved for individual users.
* Posts, comments, and todos can be created for individual users.

This service can be used to test different API calls and authentication methods.

Here, we are going to perform the API Chaining concept with the User API endpoints with the use of Token authentication.



# Authorization

**Authorization** is the process of verifying that a user or application has the necessary permissions to access a specific API resource or perform a particular action.

Postman provides an intuitive interface to manage various authorization schemes. When you send a request, Postman automatically includes the required authorization details based on the configuration you've set up.

# Generate Token

The below link is used to generate the access token to test the API.

<https://gorest.co.in/consumer/login>

You can generate the same using any of the below login mechanism.

1. Google
2. GitHub
3. Microsoft

# Using Authorization in Postman

* **Open the Authorization tab:** In the request builder, click on the "Authorization" tab.
* **Select the type:** Choose the appropriate authorization type from the dropdown.
* **Provide credentials:** Fill in the required fields based on the chosen type (e.g., username, password, token, etc.).
* **Test your request:** Send the request to verify if the authorization is working correctly.

## Example: Using Bearer Token

If your API requires a Bearer token for authentication, you would:

1. Select "Bearer Token" in the Authorization tab.
2. Paste the token in the "Token" field.
3. Send your request.

# Use of Authorization from Global/Collection/Environment Variable

* **Environment variables:** Store sensitive information like API keys or tokens in environment variables for better security.
* **Global variables:** Use global variables to share authorization details across multiple requests.
* **Collections:** Organize your requests into collections and share them with teams.