DAY 8 (22-08-25)

**API (Application Programming Interface)**

# What is an API?

* An **API**, or **Application Programming Interface**, is a set of rules and protocols that allows different software applications to communicate with each other.

Think of it as a menu in a restaurant. You, the customer, are the **client**, and the kitchen is the **server**. The menu lists the dishes you can order and what the kitchen can prepare. You don't need to know how the kitchen makes the food (the underlying code); you just need to know how to order from the menu (the API).

# How APIs work?

* APIs work using a client-server model. A client (like a web browser or a mobile app) makes a **request** to a server. This request asks for a specific piece of information or asks the server to perform a specific action.
* The request is sent in a specific format (often **JSON** or **XML**) and includes a URL endpoint, which is the specific address for the resource you're asking for.
* The server receives the request, processes it, and then sends back a **response**. The response contains the requested data or a confirmation that the action was completed.
* This response is also formatted, usually in JSON or XML, making it easy for the client to parse and use the data.

# Common types of API?

**Web APIs:** These are the most common type and are used to allow communication between a web server and a web browser. They're the backbone of most web applications and services.

**Operating System APIs:**These allow applications to interact with the operating system itself, like a program using the API to access the file system or display a window on the screen.

**Database APIs:** These let developers access and manipulate data stored in a database.

# Why Are APIs Important?

* APIs are crucial for modern software development because they promote **modularity** and **interoperability**.
* They allow developers to build new applications by leveraging the functionality of existing services without having to build everything from scratch.

For example, a travel app might use Google Maps' API to show a map, a weather API to display the current temperature, and a flight booking API to find and book flights. This makes development faster and more efficient.

# HTTPmethods

* In the context of web development, **HTTP methods** are a set of commands that define the type of action a client wants to perform on a web server's resource.
* They tell the server what to do with the resource, that is specified in the URL.

For instance, you could tell the server to "get" a user's data or "delete" a user's account.

* These methods are the foundation of how clients (like your web browser) and servers communicate in the **Hypertext Transfer Protocol (HTTP)**.
* They are an essential part of creating **REST APIs**, as they map directly to the common **CRUD** (Create, Read, Update, Delete) operations.

## Common HTTP Methods

Here are the most common HTTP methods and their typical uses:

* **GET**

This is used to **retrieve data** from a server. It's a "read-only" action.For example, when you type a URL into your browser or click a link, your browser sends a GET request to fetch the web page. It's considered a **safe** method because it doesn't change anything on the server.

* **POST**

This is used to **send data to the server to create a new resource**.For example, when you submit a form to sign up for a new account or add a new post to a blog, the data is sent via a POST request.

* **PUT**

This is used to **fully update or replace an existing resource**.If a resource with the specified ID already exists, PUT replaces it entirely with the data provided in the request body. If it doesn't exist, it can create a new resource at that location.

* **PATCH**

This method is used to **partially update an existing resource**. Unlike PUT, which requires you to send the entire resource, PATCH allows you to send only the data you want to change.For example, if you only want to update a user's email address, a PATCH request is more efficient than a PUT.

* **DELETE**

This is used to **remove a specified resource** from the server.15 For example, deleting a user's account or a specific photo.