



جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

كلية العلوم
قسم الانظمة الطبية الذكية

Lecture: (2)

Subject: Terminologies of Web Development

Level: Third stage

Lecturer: Msc najwan thaeer ali



Terminologies of Web Development

Understanding the core terminologies of web development is essential before starting with full stack development. Concepts like API, JSON, XML, and HTML form the foundation of how data is transferred, structured, and displayed in modern web applications.

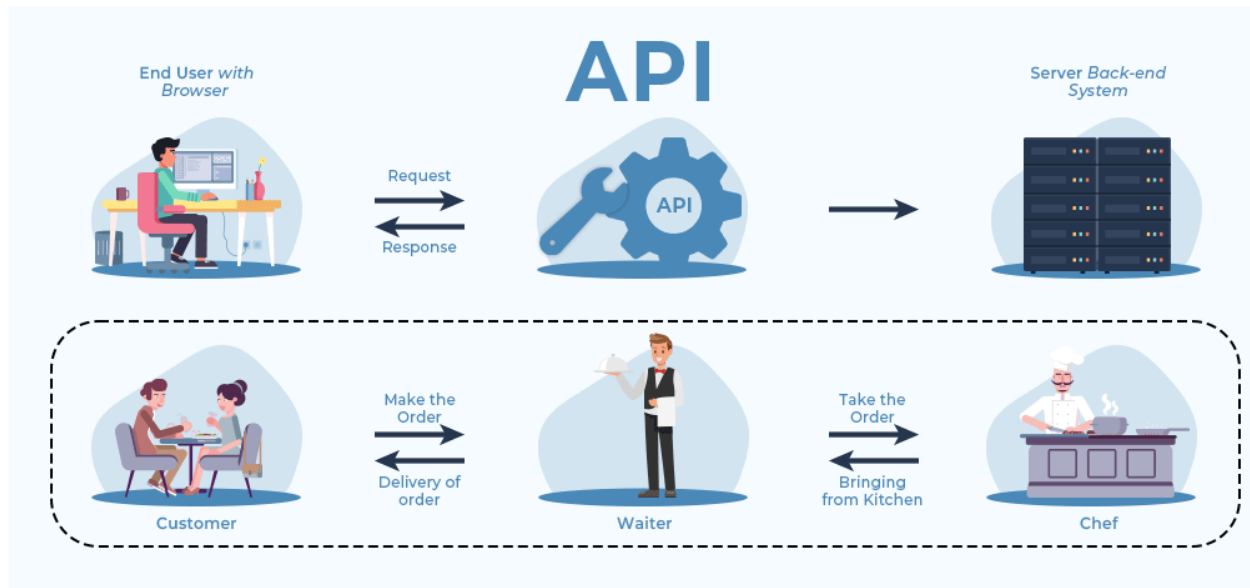
What is an API?

An API (Application Programming Interface) is like a middleman between two software systems. It allows them to talk to each other and exchange data in a structured way. It is like an Invisible Connector that makes Apps work and connects your frontend and backend.

Consider an example of using a food delivery app. You enter your address, choose your food and place the order.

- The app sends your order details to the restaurant using an API.
- The restaurant confirms and updates delivery status - again, through the API.

- You get real-time updates - thanks to APIs working in the background.



The Role of APIs in Full Stack Development

Here's how APIs fit in full stack development:

- **Fetch Live Data in the Frontend:** Use JavaScript to call APIs and bring in real-time data like user profiles, product listings or order status directly into your web app.
- **Build Powerful Backend APIs:** As a full stack developer, you will create your own APIs that handle business logic, interact with databases, and send responses to the frontend.
- **Integrate External Services:** Want to enable payments, maps, or social logins? You will connect with third-party



APIs like Stripe, Google Maps, Firebase, or Auth0 to power your app with advanced functionality

Real World Example

Let's say you are ordering food using a delivery app like Zomato or Swiggy. Here is how APIs work behind the scenes to make the experience smooth:

1. **Browse Restaurants:** You open the app and see a list of restaurants near you. The app sends your location to the server through an API, and the server returns a list of nearby restaurants.
2. **Place an Order:** You select your favorite meal and tap "Order." An API request sends your order details to the restaurant's system.
3. **Track Order Status:** The restaurant starts preparing your food and updates the order status. That status is updated in the app through another API, showing "Preparing," "Out for Delivery," etc.
4. **Live Delivery Tracking:** You watch your delivery partner approach in real time. The app uses the Google Maps API to fetch and display the live location.



Every interaction between the app, restaurant, and map service is powered by APIs — quietly sending and receiving data in the background to keep everything connected and up-to-date.

What is JSON?

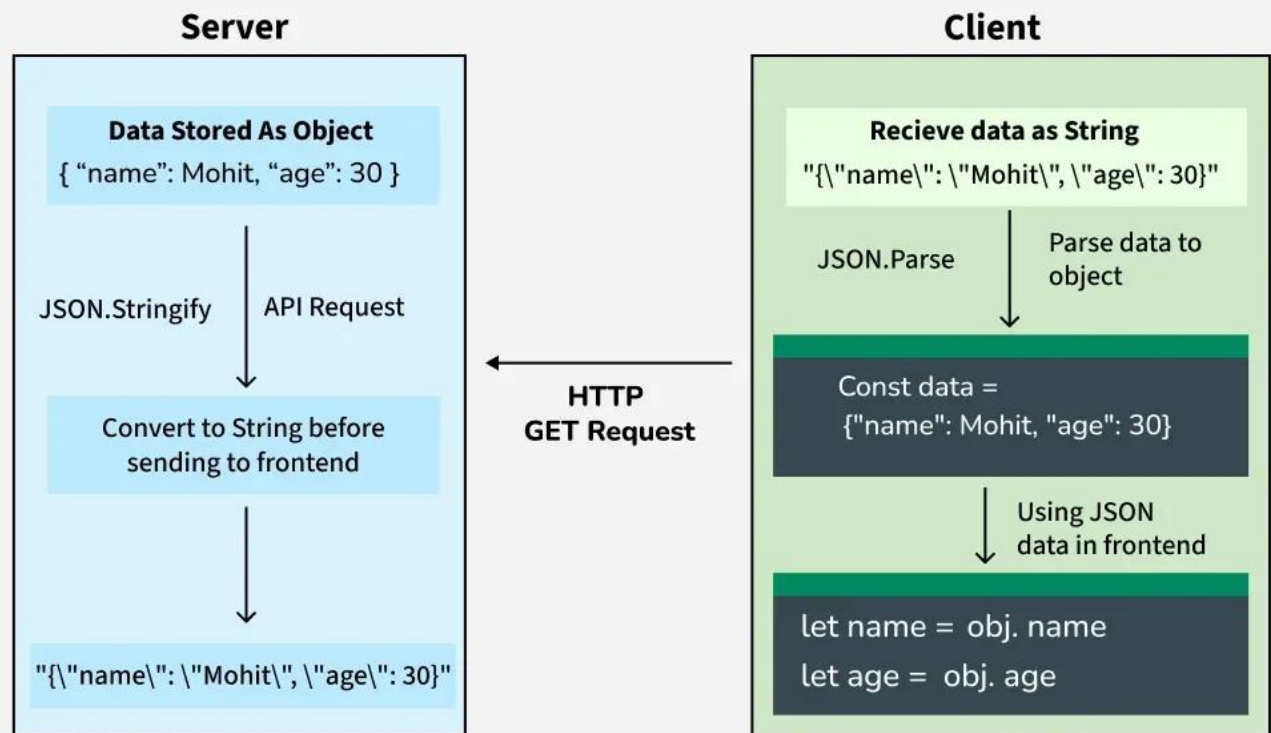
JSON (JavaScript Object Notation) is the universal format for web data exchange. It ensures that both the frontend and backend understand the information being shared between them.

- It acts as the standard format for sending and receiving data in most APIs.
- It's lightweight, easy to read, and structures data in a simple, human-friendly way

Example:

```
{  
  "name": "Sara",  
  "email": "sara@example.com",  
  "isPremiumUser": true  
}
```

- Each piece of data is a key-value pair
- Text values are inside quotes " "
- Booleans (true/false) and numbers don't need quotes
- Arrays and nested objects are also supported.



Working of JSON in Full Stack Projects

As a full stack developer, JSON will be everywhere. Here is how:

- **Receive Data from APIs:** Get user profiles, product lists, orders, etc., all in JSON format.
- **Send Data to the Backend:** Forms, login info, and checkout details are often submitted as JSON.
- **Store Data in Frontend Apps:** Keep session info, cart data, or UI settings in JSON inside local storage or state.

Difference between HTML and XML



Both [HTML](#) and [XML](#) look similar, but they serve very different purposes. Both use tags like <this>, but their roles in web development are completely different. Given below are the key differences between XML and HTML:

XML	HTML
XML stands for Extensible Markup Language.	HTML stands for HyperText Markup Language.
Used to store and transport data.	Used to display data in a web browser.
User-defined tags (e.g., <book>, <price>).	Predefined tags (e.g., <h1>, <p>, <div>).
Case-sensitive — <Name> ≠ <name>.	Not case-sensitive — <TITLE> = <title>.
Requires strict syntax — every tag must be closed properly.	Flexible syntax — missing tags are often tolerated.
Does not display content	Designed for visual



XML	HTML
visually.	presentation.
Tags are used to describe data.	Tags are used to display data.
No predefined structure — very flexible.	Follows a fixed structure for rendering.

Note:

- *HTML is for browsers.*
- *XML (like JSON) is for how systems communicate.*
- *JSON is now the most common choice for modern web apps.*