In Android development, a dynamic app refers to an application where certain components or functionalities are loaded and executed at runtime, rather than being entirely predefined and packaged with the initial installation of the app. Dynamic apps often leverage features such as dynamic code loading, dynamic module loading, or remote content loading to provide flexibility and adaptability.

Here's a comparison between static and dynamic apps in tabular form:

Feature	Static App	Dynamic App
Loading of Content	All content and functionalities are packaged	Certain components or functionalities are loaded
	with the initial installation of the app.	and executed at runtime.
Flexibility	Less flexible because changes or updates	More flexible as updates or changes to
	require app updates and redistribution	certain components can be done without
		distributing a new version of the app.
App Size	Generally larger since all components are	Potentially smaller initial app size as some
	bundled with the installation package.	components may be downloaded or loaded
		dynamically.
Distribution	Single package distributed through app stores	May require additional server infrastructure
	or other means.	for dynamic content delivery.
Updates	Updates require users to download and install	Updates can be pushed dynamically, potentially
	a new version of the app.	reducing user effort and improving
		time-to-update.
Offline Capability	Generally better offline capability since	Offline capability might be limited depending
	all components are pre-installed.	on the app's design and the availability
		of dynamically loaded content.
Examples	Traditional standalone apps, games	Apps with modules that can be downloaded
		from servers, apps with dynamic UI elements,
		apps with remote configurations or features.

Overall, dynamic apps offer more flexibility and can provide a better user experience in terms of updates and adaptability. However, they may require more complex development and infrastructure management compared to static apps.

What is Dynamic

- (of a process or system) characterized by constant change, activity, or progress.

- Similarly a Dynamic App's Content, design or functionality changes w.r.t Time.

- For eg: Whatsapp, Facebook,..etc

What is API (REST)

- REST determines how the API looks like. It stands for “Representational State Transfer”.

- Each URL is called a request while the data sent back to you is called a response.

What is API (REST)

- An API is an Application Programming Interface.

- It is a set of rules that allow programs to talk to each other.

- The developer creates the API on the server and allows the client to talk to it.

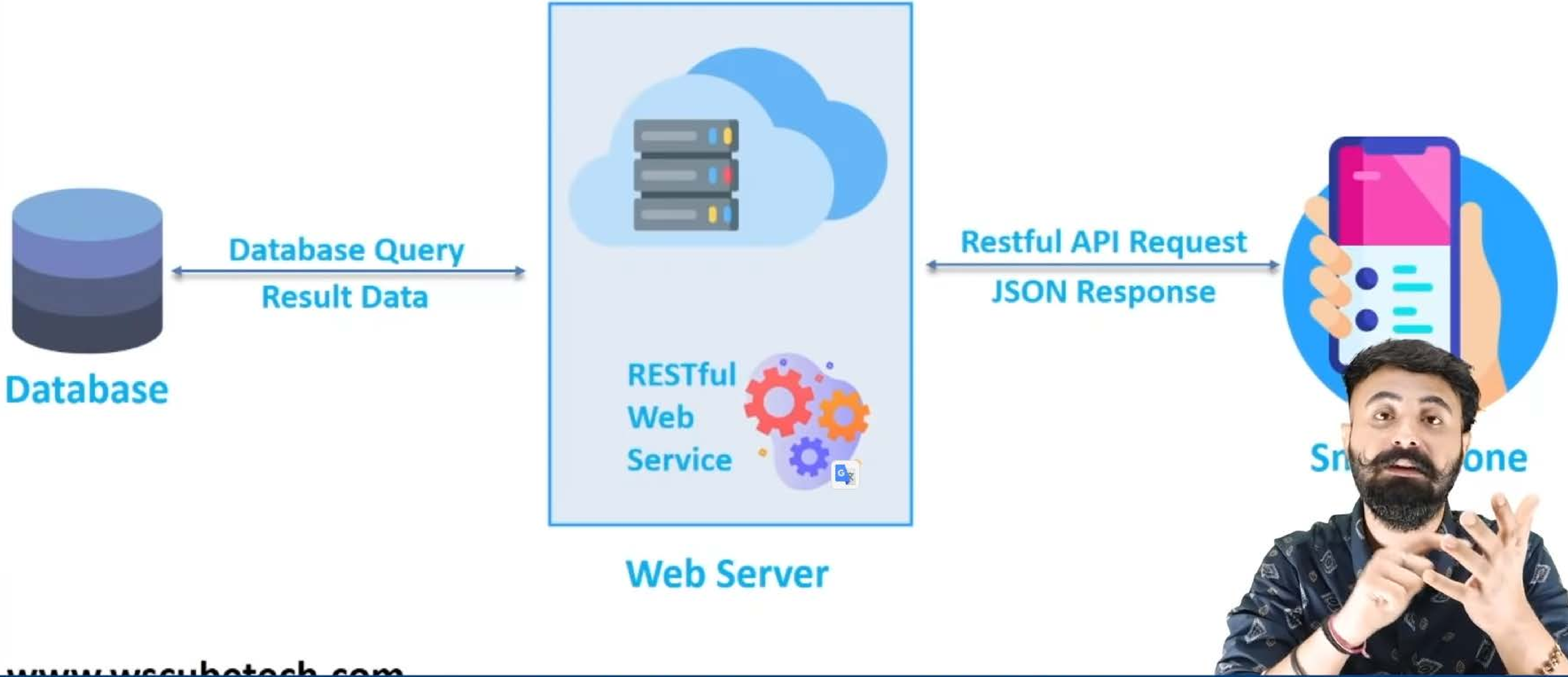
- ✓ GET API: GET API ek prakar ka API hai jo data retrieve karne ke liye use kiya jata hai. Isme hum kisi bhi resource se data mangte hain aur woh humare request ke saath URL ke through return hota hai. Iske kuch examples hain: fetching user details, getting a list of items, etc.

- ✓ POST API: POST API ek aur prakar ka API hai jo data update or create karne ke liye use kiya jata hai. Isme hum kisi bhi resource mein data send karte hain aur woh humare request ke saath HTTP request body ke through return hota hai. Iske kuch examples hain: creating a new user, updating user details, etc.

In dono API mein se koi bhi HTTP request method use kiya jata hai, jaise ki GET, POST, PUT, DELETE, etc. Har request method ke apne functionality hai aur inko alag-alag scenarios mein use kiya jata hai.

Mujhe ummeed hai ki yeh jaankari aapke liye helpful rahi hogi! Koi bhi sawal ho toh mujhe bataiye, main aapki madad karna pasand karungi.

Behind the scenes of Dynamic App



Database

Database Query

Result Data

RESTful Web Service

Web Server

Restful API Request

JSON Response

Sn one

What is JSON

- JSON stands for JavaScript Object Notation
- JSON is a lightweight format for storing and transporting data
- JSON is often used when data is sent from a server to a web page
- JSON is "self-describing" and easy to understand

JSON Rules

- Data is in name/value pairs
- Data is separated by commas
- Curly braces hold objects { }
- Square brackets hold arrays [ ]

JSON Data

- JSON data is written as name/value pairs, just like JavaScript object properties.

- A name/value pair consists of a field name (in double quotes), followed by a colon, followed by a value:

```
"firstName":"John"
```

JSON Objects

- JSON objects are written inside curly braces.
- Just like in JavaScript, objects can contain multiple name/value pairs:

```
{"firstName":"John", "lastName":“Harper“}
```

JSON Arrays

- JSON arrays are written inside square brackets.

- Just like in JavaScript, an array can contain objects:

```
"Students":[
   {"firstName":"John", "lastName":"Harper"},
   {"firstName":"Anna", "lastName":"Jones"}
   {"firstName":“Bret", "lastName":"Lee"}
]
```