**Application Programming Interface**

* It's a way for different software applications to communicate with each other.
* APIs define rules that allow one program to send requests to another and get responses.
* They enable data exchange between systems (e.g., between a website and a database).
* APIs can be used to access the functionality of other services or platforms (e.g., using Google Maps on your website).
* There are different types of APIs, like REST (web-based) and GraphQL.
* APIs help simplify development, allowing you to integrate features without building them from scratch.
* They often require authentication to control access, ensuring only authorized users can use them.

**“The cooks in a kitchen are the back end, the dining area is the front end, and the servers are the APIs.”**

A REST API (Representational State Transfer API) is a type of web-based API that follows a set of principles for building and interacting with web services.

* REST is an architectural style that uses HTTP for communication.
* REST APIs typically involve request-response interactions between a client and a server.
* Clients make requests to access or manipulate resources (like data) on the server, usually via URLs.
* RESTful APIs use standard HTTP methods like:
* GET – to retrieve data (get user details)
* POST – to send or add data (submit a form)
* PUT – to update existing data (edit a post)
* DELETE – to remove data (delete an account)
* Responses are often sent in JSON format, making them easy to read and work with in modern web applications.
* REST APIs are stateless, meaning each request from a client contains all the information needed to process the request, without relying on previous interactions.
* They are scalable and commonly used in web and mobile apps to connect to backend services or databases.
* Versioning is important as it allows an implementation to provide backwards compatibility (This prevents prior versions from breaking in an update)

**Examples**

GET /api/users retrieves a list of users.

POST /api/users creates a new user.

Queries can return codes

* 200 level (Success)
* 400 level (Something went wrong with our request)
* 500 level (Something went wrong at our server level)

Workflows can include

1. Client makes an HTTP request
2. API receives the request
3. API interacts with PostgreSQL
4. PostgreSQL processes query
5. Database returns the result
6. API sends a response to the client