LinkedIn article about RESTful API

What is a REST API?

REST API stands for **RE**presentational **S**tate **T**ransfer **API**. It is a type of **API** that allows communication between different systems over the internet. **REST API** work by sending requests and receiving responses, typically in **JSON format**, between the client and server.

History of REST APIs

- **Before REST**, developers used SOAP, manually writing XML documents and sending RPCs.

- **2000**: Roy Fielding and a group of developers defined REST standards to simplify server communication**.**

- **2002**: eBay and Amazon launched their REST APIs, expanding their reach.

- **2004–2006** : Flickr, Facebook, and Twitter released their APIs to better support developers and prevent unofficial data scraping**.**

- **2006-Now** :RESTful APIs became the standard for adding functionality to websites and apps, with tools like Postman making API development faster and easier .with a Remote Procedure Call (RPC) in the body.

The Six Guiding Principles of REST

1. **Use of a uniform interface** (UI)
   To have a uniform interface, multiple architectural constraints are required to guide the behavior of components. Additionally, resources should be unique so they are identifiable through a single URL.

2. **Client-server based**
   The uniform interface separates user concerns from data storage concerns. The client's domain concerns UI and request-gathering, while the server's domain concerns focus on data access, workload management, and security. The separation of

client and server enables each to be developed and enhanced independently of the other.

3. **Stateless operations**
   Request from client to server must contain all of the information necessary so that the server can understand and process it.

4. **RESTful resource caching**
   Data within a response to a request must be labeled as cacheable or non-cacheable.

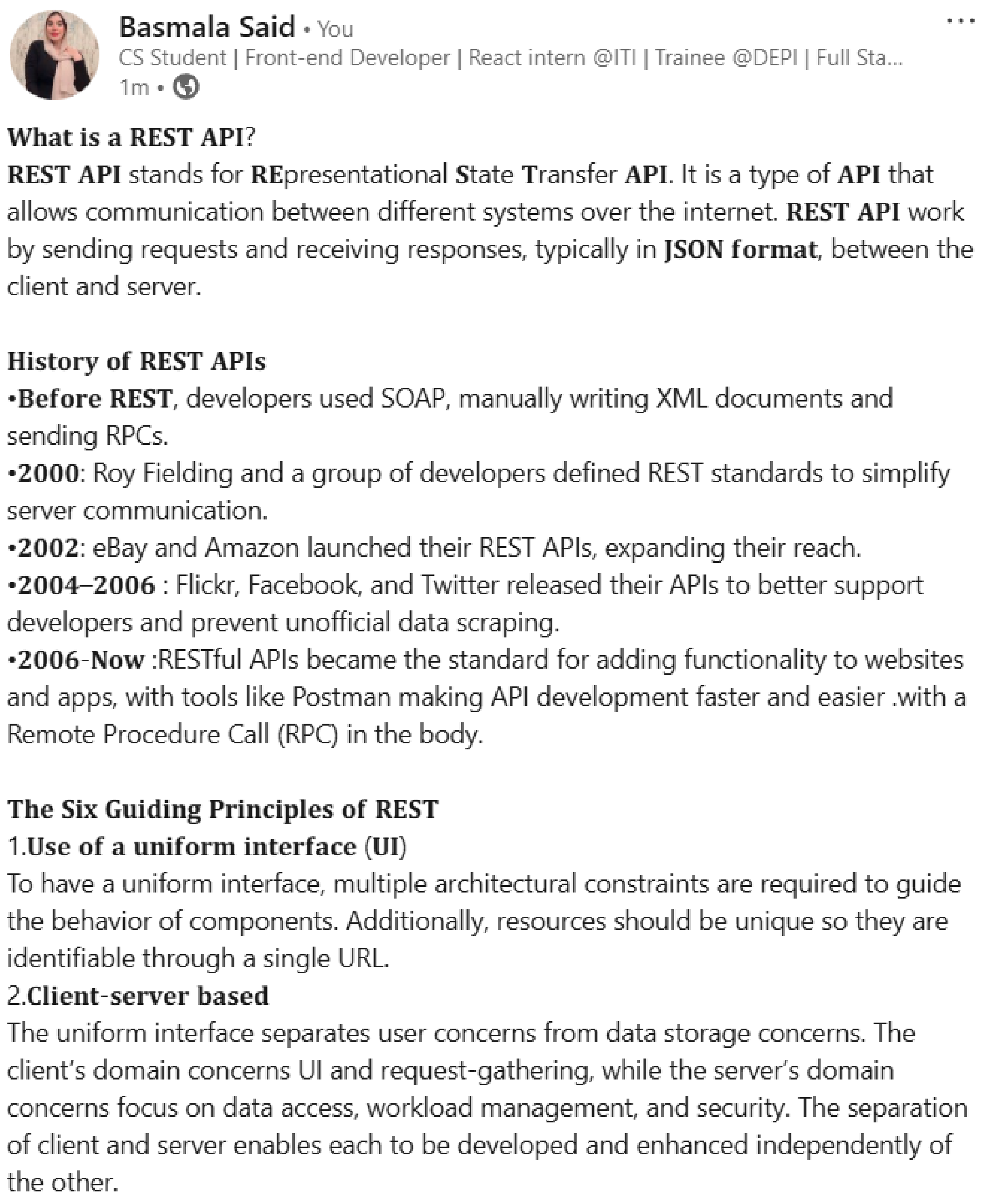
5. **Layered system**
   REST allows for an architecture composed of hierarchical layers. In doing so, each component cannot see beyond the immediate layer with which they are interacting.

6. **Code on demand**
   Because REST APIs download and execute code in the form of applets or scripts, there's more client functionality. Oftentimes, a server will send back a static representation of resources in the form of **XML or JSON**. Servers can also send executable codes to the client when necessary.

The four main resource methods that are associated with REST APIs are:

1. **GET**: This method allows for the server to find the data you requested and sends it back to you.
2. **PUT**: If you perform the 'PUT' request, then the server will update an entry in the database.
3. **POST**: This method permits the server to create a new entry in the database.
4. **DELETE**: This method allows the server to delete an entry in the database.

More information about the **REST APIs** (https://blog.postman.com/rest-api-examples/)



Basmala Said • You

CS Student | Front-end Developer | React intern @ITI | Trainee @DEPI | Full Sta...

1m •

What is a REST API?

REST API stands for **RE**presentational **S**tate **T**ransfer **API**. It is a type of **API** that allows communication between different systems over the internet. **REST API** work by sending requests and receiving responses, typically in **JSON format**, between the client and server.

History of REST APIs

•**Before REST**, developers used SOAP, manually writing XML documents and sending RPCs.

•**2000**: Roy Fielding and a group of developers defined REST standards to simplify server communication.

•**2002**: eBay and Amazon launched their REST APIs, expanding their reach.

•**2004–2006** : Flickr, Facebook, and Twitter released their APIs to better support developers and prevent unofficial data scraping.

•**2006-Now** :RESTful APIs became the standard for adding functionality to websites and apps, with tools like Postman making API development faster and easier .with a Remote Procedure Call (RPC) in the body.

The Six Guiding Principles of REST

1.**Use of a uniform interface** (**UI**)

To have a uniform interface, multiple architectural constraints are required to guide the behavior of components. Additionally, resources should be unique so they are identifiable through a single URL.

2.**Client-server based**

The uniform interface separates user concerns from data storage concerns. The client's domain concerns UI and request-gathering, while the server's domain concerns focus on data access, workload management, and security. The separation of client and server enables each to be developed and enhanced independently of the other.

3.**Stateless operations**

Request from client to server must contain all of the information necessary so that the server can understand and process it.

4.**RESTful resource caching**

Data within a response to a request must be labeled as cacheable or non-cacheable.

5.**Layered system**

REST allows for an architecture composed of hierarchical layers. In doing so, each component cannot see beyond the immediate layer with which they are interacting.

6.**Code on demand**

Because REST APIs download and execute code in the form of applets or scripts, there's more client functionality. Oftentimes, a server will send back a static representation of resources in the form of **XML or JSON**. Servers can also send executable codes to the client when necessary.

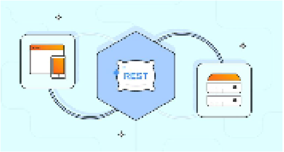
The four main resource methods that are associated with REST APIs are:

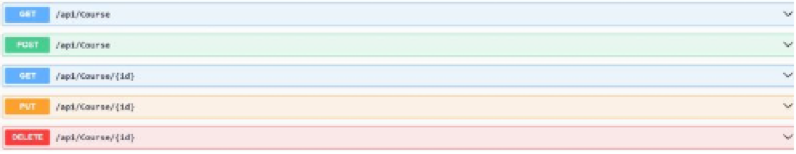
1.**GET**: This method allows for the server to find the data you requested and sends it back to you.

2.**PUT**: If you perform the 'PUT' request, then the server will update an entry in the database.

3.**POST**: This method permits the server to create a new entry in the database.

4.**DELETE**: This method allows the server to delete an entry in the database.





GET /api/Course

POST /api/Course

GET /api/Course/{id}

PUT /api/Course/{id}

DELETE /api/Course/{id}