What is rest app ?

A **RESTful API** follows the principles of **Representational State Transfer (REST)** to ensure scalability, simplicity, and efficiency. To be considered truly RESTful, an API should follow these **key constraints**:

### 1. ****Client-Server Architecture****

* The client and server are separate, allowing independent evolution.
* The client sends requests, and the server processes them and returns responses.

### 2. ****Statelessness****

* Each request from the client contains all the information needed to process it.
* The server does not store client session data between requests.

### 3. ****Uniform Interface****

* The API should have a consistent and predictable structure.
* Uses **standard HTTP methods**:
  + GET → Retrieve data
  + POST → Create a new resource
  + PUT → Update an existing resource
  + PATCH → Partially update a resource
  + DELETE → Remove a resource
* Uses **resource-based URIs**:
  + ✅ /users/123 (Good)
  + ❌ /getUser?id=123 (Bad)
* Data should be represented in standard formats like **JSON** or **XML**.

### 4. ****Resource-Based Approach****

* Everything is treated as a resource (/users, /orders, /products).
* Resources are represented using **URIs** (Uniform Resource Identifiers).

### 5. ****Layered System****

* The API should allow for multiple layers (e.g., authentication, caching, load balancing) without the client knowing the details.

### 6. ****Cacheability****

* Responses should include caching headers to reduce redundant requests.
* Example: Cache-Control: max-age=3600 (tells clients to cache the response for an hour).

### 7. ****Code on Demand (Optional)****

* The server can send executable code (e.g., JavaScript) to the client if needed.

### ****Best Practices for a RESTful API****

✔ Use **plural nouns** for resources (/users instead of /user).  
✔ Implement **proper status codes** (200 OK, 201 Created, 400 Bad Request, 404 Not Found).  
✔ Use **HATEOAS** (Hypermedia as the Engine of Application State) if needed, linking related resources.  
✔ Secure the API with **authentication** (OAuth, JWT, API keys).

Would you like an example RESTful API structure? 🚀

Query Paramter

http://localhost:3000/filter?type=Puns