What is Representational State Transfer (REST)?

REST is a set of **constraints** and **principles** that define how web standards like **HTTP** should be used to create **web services**. It was introduced by **Roy Fielding** in his 2000 doctoral dissertation.

Key Concepts:

Concept	Description
Resource	Everything in REST is a **resource** (e.g., a user, a file, an order) identified by a **URI** (Uniform Resource Identifier).
Representation	A **representation** is how a resource is transferred — usually in formats like **JSON**, **XML**, or **HTML**.
Stateless	Each client request must contain **all the information** needed for the server to understand and process it — no session or memory is stored on the server between requests.
HTTP Methods	Standard HTTP verbs are used: GET, POST, PUT, DELETE, etc., each representing an action.
Client-Server	REST separates the user interface from the data storage, improving scalability and flexibility.
Cacheable	Responses must explicitly state whether they are cacheable to improve performance.
Uniform Interface	A consistent, simple interface across the API using standard HTTP protocols and naming conventions.

"Representational" in REST

- The term **"representational"** means that the **server sends a representation** (usually a JSON or XML document) of the resource.
- For example, if you request GET /users/123, the server might return:

```
{
  "id": 123,
  "name": "Shekhar",
  "email": "shekjava@gmail.com"
}
```

This **representation** shows the **state** of the user resource at the time of the request.

RESTful API Example

HTTP Method	URI	Description
GET	/books	Get all books
GET	/books/10	Get book with ID 10
POST	/books	Create a new book
PUT	/books/10	Update book with ID 10
DELETE	/books/10	Delete book with ID 10

- **"Representational State Transfer"** = Transferring **representations** of **resource state** between client and server using **standard HTTP methods**.
- RESTful systems are **stateless**, **cacheable**, and use a **uniform interface**.