

REST API

-CSED, TIET

What is REST API

- REST: Representational State Transfer
- It is an architectural style that defines a set of rules to create web service.
- Client Server communication.
- API creates an object of a particular request

REpresentation State Transfer

1. **Representation :** Data or resources are encoded as representations of data or resource.
2. **State:** All the necessary state for the request must be provided with the request.
3. **Transfer:** The representations and the state can be transferred between client and server.

Principles of REST

- Stateless
- Client Server
- Uniform interface
- Cacheable
- Layered system

Stateless

- The server does not store any session data
- The communication between the client and the server is stateless
- All the information to understand a request is contained within the request
- It improves scalability
- Requires more bandwidth

Client Server

- REST application should have a **client server architecture**
- Client and server are **separated**
- They can evolve independently
- Clients need not know anything about business logic
- Servers need not know anything about frontend UI

Uniform Interface

- Uniform interface is the main advantage of REST APIs
- Different types of devices should have same way of interacting with server.
- There are 4 elements of Uniform interface
 - Identification of resources
 - Manipulation of resources through representation
 - Self-descriptive messages for each request
 - HATEOS (Hypermedia As The Engine of Application State)

Cacheable

- Responses should be cacheable if possible
- For subsequent request, the client can retrieve from its cache, need not send the request to the server
- Network optimization is improved
- Users can get their information

Layered System

- Allows an architecture to be composed of hierarchical layers
- Each layer doesn't know anything beyond immediate layer
- Limits the amount of complexity that can be introduced at any single layer
- Advantage: Security

Methods of REST API

CRUD operations

1. C : Create -> POST
2. R: Read -> GET
3. U: Update -> PUT
4. D: Delete -> DELETE



HTTP Methods

Key Elements

- Resources
- Request Verbs
- Request Headers
- Request Body
- Response Body
- Response Status Codes

Thank You