REST – Representational State Transfer

1. **What is REST?**
   REST are the set of rules that are used to design the architecture between a system and the web.REST makes communication between two system easier.

   1. **Independent Implementation**
      The REST architecture allows the independent implementation of the client and the server without each knowing about the other.It means that the code on the client side can be changed at any time without affecting the operation of the server and the code on the server side can be changed without affecting the operation of the client.
   2. **Stateless**
      Each request from client should have all the necessary information in it and that should not use any other information available at the server.

   3. **Message Format**
      In the REST server and client should know the format of message to be send to others.
      1. **XML**
         In the REST the simple http request is sent to server and then as a response from a server the XML document is sent.REST focuses more on the resources the URL ha sbeen given to the resource.the URL for a resource can be-
         http://social.john.com/profile/johnjose

         The above URL can return the following XML document.

```
<profile>
        <firstName>john</firstName>
        <lastName>Jose</lastName>
        <address>
           <street>The Road 123</street>
           <zip>123457</zip>
           <city>India</city>
        </address>
</Profile>
```

      2. **JSON**
         As a response simple JSON document is returned for the URL.The advantage of JSON over XML is that web browser are able to parse the JSON structures into JavaScript objects.

```
{
      firstName : "John",
      lastName  : "Jose",
      address   : {
      street : "The Road 123",
      zip    : "12345",
      city   : “India"
}

}
```

2. Making Request

The REST requres client to send request to the server in order to retrieve or modify the data on server.A request consist following-

1. HTTP Methods

There are four HTTP methods that we use in requests to interact with resources in a REST system

1. **GET** - GET requests retrieve resource information only and not to modify it also GET does not change the state of resource therefore it is called safe method. When GET retrieves any resource information successfully it return 200(OK) status code.In case resource is NOT found on server then it must return HTTP response code 404(NOT FOUND).

2. **POST** - POST methods are used to create a new resource into the collection of resources.if a resource has been created on the origin server, the response should be HTTP response code 201(CREATED),and in case if the methode could not identify which resource to serve it return status code 204(NO CONTENT).

3. **PUT** - PUT primarily used to update existing resource. If a new resource has been created by the PUT, the origin server must inform the user agent via the HTTP response code 201 (CREATED) response and if an existing resource is modified, either the 200 (OK) or 204 (No Content) response codes should be sent to indicate successful completion of the request.

4. **DELETE** DELETE method used to Delete the resource.

2. Headers

The header of the request includes the field called **accept** that contains the type of contents client wants from the server as a response.content type of accept field consists of type and subtype those are separated by slash(/).

Consider,If a client wants to receive simple html file as a response it will add its contents type in accept field as *text/html*.

1. Content types and Subtypes

1. image - image/png, image/jpeg, image/gif

2. audio - audio/wav

3. video - video/mp4, video/ogg

3. Resource Path

Request must contain a pathe to the resources so that client can get to know that what is he working on.A path like fashion.com/customers/12/orders/4 can be get to know by seeing

only as its following the hyrarchy of operations that are going performed on resources ,we can get that we are accessing the order having id 4 and customer whose id is 12.

3. Sending Response

When server is sending the data to the client it should mention the content type that it is going to send so that the client will get to know that what type of content is that.In case that the client want that data the same content type should be available in its accept field of request.

Request-

GET/article/23

Accept: text/html

Response-

HTTP/1.1 200(ok)

content-type : text/html

As the content-type of server includes text/html client will able to accept that.

4. Status Codes

Responses from the server contain status codes to alert the client to information about the success of the operation. The most common ones and how they are used are as follows-

1. **200(OK)**

   This is the standard response for successful HTTP requests.

2. **201(CREATED)**

   This is the standard response for an HTTP request that resulted in an item being successfully created.

3. **204(NO CONTENT)**

   This is the standard response for successful HTTP requests, where nothing is being returned in the response body.

4. **400 (BAD REQUEST)**

   The request cannot be processed because of some bad request syntax, excessive size, or another client error.

5. **403(FORBIDDEN)**

   The client does not have permission to access this resource**.**

6. **404(NOT FOUND)**

   The resource could not be found at this time. It is possible it was deleted, or does not exist yet

7. **500(INTERNAL SERVER ERROR)**

   In case any unexpected error occurs with server and have no specific information available then this code is shown.

- Each HTTP method returns a status code upon success-
  - GET - Return 200(OK)
  - POST - Return 201(CREATED)
  - PUT - Return 200(OK)
  - DELETE - Return 204(NO CONTENT)