* HTTP protocol is independent of any language like JAVA etc
* End point: Address where API is hosted on the Server.

HTTP methods which are commonly used to communicate with Rest API’s are -

**GET, POST, PUT, and DELETE – CRUD Operations**

GET- The GET method is used to extract information from the given server using a given URI. While using GET request, it should only extract data and should have no other effect on the data. No Payload/Body required

**How to send input data in GET?**  
Ans: Using Query Parameters

POST- A POST request is used to send data to the server, for example, customer information, file upload, etc. using HTML forms.

**How to send input data in POST?**  
Ans: Using Form Parameters /Body Payload

PUT- Replaces all current representations of the target resource with the uploaded content.

DELETE- Removes all current representations of the target resource given by a URI.

**Resources:**Resources represent API/Collection which can be accessed from the Server

Google.com/maps  
google.com/search  
google.com/images

**Path Parameters:**  
***Path parameters*** are variable parts of a URL path. They are typically used to point to a specific resource within a collection, such as a user identified by ID

<https://www.google.com/Images/1123343>  
<https://www.google.com/docs/1123343>  
<https://amazon.com/orders/112>

<https://www.google.com/search?q=newyork&oq=newyork&aqs=chrome..69i57j0l7.2501j0j7&sourceid=chrome&ie=UTF-8>

**Query Parameters:**  
Query Parameter is used to sort/filter the resources.

Query Parameters are identified with?””

https://amazon.com/orders?sort\_by=2/20/2020

**Headers/Cookies**:

Headers represent the meta-data associated with the API request and response. In layman terms, we bwere sending Additional details to API to process our request.  
Example: Authorization details.

**EndPoint -** Base URL+ Resource+Parameters(Query+/Path)

**End Point Request URL can be constructed as below**  
Base URL/resource/(Query/Path) Parameters

* To store the response of the http call as string, we use – extract().response().asString() Method
* JsonPath – it is a class, object of this class, takes string as input and converts it into Json.
* Session Filter – it is an class used in restAssured, to remember the authentication and cookie of one session to other subsequent sessions.
* To bypass the HTTP certification, HTTP authentications while interacting with server, we can use – **relaxedHttpValidation()**

There are multiple types of authorisations for example =

1. Cookie based authorisation.
2. Oauth 2.0
3. Basic key authorisation

**Oauth2.0 –** comes with multiple grant type. Most used 2 types are

1. Authorisation code
2. Client Credentials

**Oauth 2.0 Mechanism (Authorisation Code) -**

1. **Client –** it is the customer who needs to be served. (Book my show)
2. **Client ID –** ID the that identifies the client. It will confirm that book my show has access to google authorisation server.
3. **Client Secret ID –** ID which client will get after registration with third party which is google.
4. **Resource Owner –** human
5. **Resource/Authorisation Server –** Google

In authorisation Code authentication method, when client wants to be registered/login/authorised using authorisation server (third party tool like google), In backend, these servers are being hit.

**1st** **Step**– user clicks on “sign in with google” on client (book my show) UI and then if client is registered with authorisation server, then using client ID user hits the authorisation server and confirms that the user is the existing customer of the authorisation server for example (user is existing customer of google)

If user is existing customer, then authorisation server will return a code (OTP) to the client.

**2nd** **Step**– Now client will hit again the resource server of the authorisation member (which google for example) and will get the resource’s details like first name, last name DOB etc. and along with that **resource server will also return the Access token** to the client.

Using access token, client will identify the user and will provide the access for there own server. Access token will be expired after some time (if customer logs out, or some error is there) then again, user need to login and get another Access token to have the access again for the client server (book my show).

**Query Parameters of Authorisation code –**

* Scope – Scope what all details will be provided by google to client about username. f name, l name.
* Auth URL – Authorisation server URL.
* Response type – which type of data you need from authorisation server i.e. code (for otp code).
* Redirect URL – after authorisation from authorisation server and resource server, on which end point of client, user needs to be redirected
* State – used for security to avoid hacking.

**#.** RestAssured by default, does encoding for few strings value like % and converts the value in numeric value. To avoid this scenario, we use one method in given as – **urlEncodingEnabled(false)**

**Client Credentials –**

**Example –** I need all twits to be shown on my browser.

**Serialisation and Deserialization -**

**Serialisation** – In restAssured context, serilisation is the process of converting java object into request body (Payload).

**Deserialsation** - – In restAssured context, deserilisation is the process of converting request body (Payload) into Java object.