**Rest assured** is java library for **testing** Restful Web services. It can be used to **test** XML & JSON based web services. It supports GET, POST, PUT, PATCH, DELETE, OPTIONS and HEAD requests and can be used to validate and verify the response of these requests.

**API Basics and Terminologies**

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**

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**:

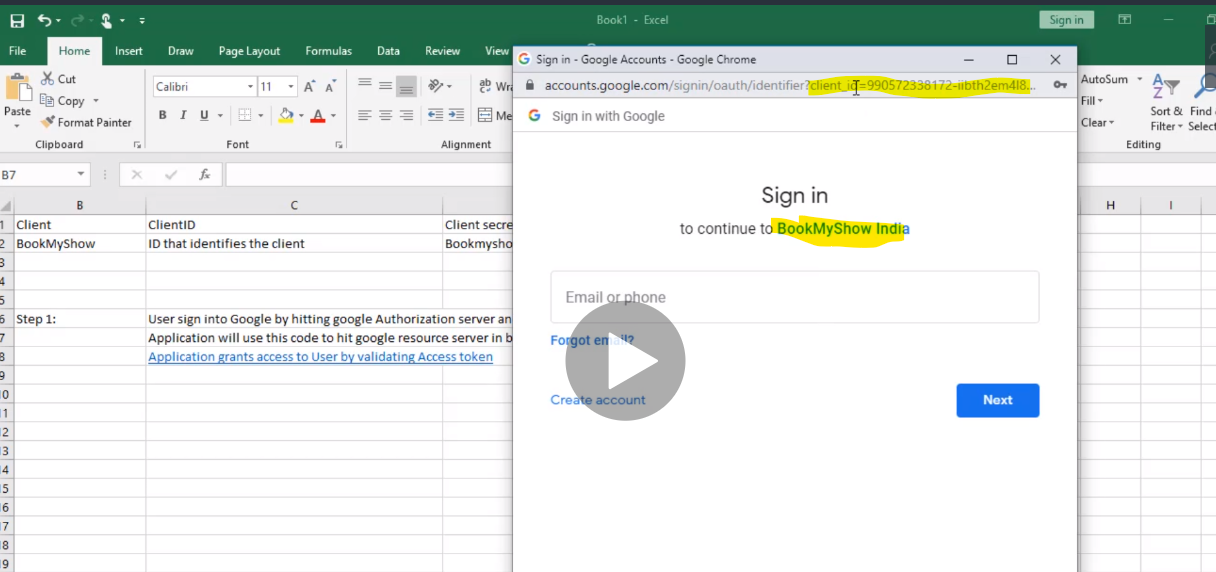
Headers represent the meta-data associated with the API request and response. In layman terms, we were sending Additional details to API to process our request.  
Example : Authorization details

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

**OAuth: section 9 very important**

There is more than one type (Authorization code, client credentials)

We can know every type need what from Postman

1. Client : website
2. Client ID (public): id that identify the client can get it from Authorization Server: Google  
   
3. Client secret ID :
4. Resource owner
5. Resource/Authorization Server: Google..Facebook..salesforce
6. After step 5 .. the resource server will give us (access token, first name, last name , email …)
7. Then this access token is stored in the website for the user

|  |  |
| --- | --- |
| GrantType | Authorization code |
| redirect URL/Callback URL | https://rahulshettyacademy.com/getCourse.php |
| Authorization server url | https://accounts.google.com/o/oauth2/v2/auth |
| Access token url | https://www.googleapis.com/oauth2/v4/token |
| Client ID | 692183103107-p0m7ent2hk7suguv4vq22hjcfhcr43pj.apps.googleusercontent.com |
| Client Secret | erZOWM9g3UtwNRj340YYaK\_W |
| Scope | https://www.googleapis.com/auth/userinfo.email |
| State | Any random string |
| How to pass oauth in request | Headers |

**Mandatory fields for GetAuthorization Code Request ;**

**End Point** : Authorization server url  
**Query Params**:Scope, Auth\_url, client\_id, response\_type, redirect\_uri

This operation should perform on browser  
output : Code

**Mandatory fields for GetAccessToken Request :  
End point** : Access token url

**Query Params** :Code, client\_id, client\_secret, redirect\_uri, grant\_type

Output : Access token

**Serialization & deserialization**

Serialization : Convert request from object to json data ( create classes same as request json)

Serialization : Convert response from json to object ( create classes same as response json)

**Request Spec Builder**

If u have common steps like queryParam or header or any baseURl for multi Requests we can use Request Spec Builder to reduce duplicate code