

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](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

---

#### **Google Maps Add API (POST):**

This API Will add new place into Server

**Complete URL :** <https://rahulshettyacademy.com/maps/api/place/add/json?key=qaclick123>

**Base URL:** <https://rahulshettyacademy.com>

**Resource:** /maps/api/place/add/json

**Query Parameters:** key =qaclick123

**Http Method:** POST

**Sample Body :**

{

```
"location": {  
  "lat": -38.383494,  
  "lng": 33.427362  
},  
"accuracy": 50,  
"name": "Frontline house",  
"phone_number": "(+91) 983 893 3937",  
"address": "29, side layout, cohen 09",  
"types": [  
  "shoe park",  
  "shop"  
],  
"website": "http://google.com",  
"language": "French-IN"  
}
```

### Sample Response

```
{  
  "status": "OK",  
  "place_id": "928b51f64aed18713b0d164d9be8d67f",  
  "scope": "APP",  
  "reference": "736f3c9bec384af62a184a1936d42bb0736f3c9bec384af62a184a1936d42bb0",  
  "id": "736f3c9bec384af62a184a1936d42bb0"  
}
```

### Google Maps Delete API (POST):

This API Will delete existing place from Server

**Complete URL:** <https://rahulshettyacademy.com/maps/api/place/delete/json?key=qaclick123>

**Base URL:** <https://rahulshettyacademy.com>

**Resource:** /maps/api/place/delete/json

**Query Parameters:** key

**Http request :** DELETE

#### Sample Body :

```
{  
  "place_id": "928b51f64aed18713b0d164d9be8d67f"  
}
```

#### Sample Response

```
{  
  "status": "OK"  
}
```

### Google Maps get Place API (GET):

**This API Will get existing place details from Server**

**Complete URL :**

[http://rahulshettyacademy.com/maps/api/place/get/json?place\\_id=xxxx&key=qaclick123](http://rahulshettyacademy.com/maps/api/place/get/json?place_id=xxxx&key=qaclick123)

**Base URL:** <https://rahulshettyacademy.com>

**Resource:** /maps/api/place/get/json

**Query Parameters:** key, place\_id //( place\_id value comes from Add place response)

**Http request:** GET

Note: Key value is hardcoded and it is always qaclick123

### Sample Response for the Provided Place\_Id

```
{
  "location":{
    "lat" : -38.383494,
    "lng" : 33.427362
  },
  "accuracy":50,
  "name":"Frontline house",

  "phone_number":"+91 983 893 3937",
  "address" : "29, side layout, cohen 09",
  "types": ["shoe park","shop"],
  "website" : "http://google.com",
  "language" : "French-IN"
}
```

### Google Maps Put Place API (PUT):

This API Will update existing place in Server with new values

#### Complete URL :

[http://rahulshettyacademy.com/maps/api/place/get/json?place\\_id=xxxx&key=qaclick123](http://rahulshettyacademy.com/maps/api/place/get/json?place_id=xxxx&key=qaclick123)

**Base URL :** <https://rahulshettyacademy.com>

**Resource:** /maps/api/place/update/json

**Query Parameters:** key

**Http Method:** PUT -

Note: Key value is hardcoded and it is always qaclick123

### Sample Request:

```
{  
  "place_id": "8d2573bdf6ceec0e474c5f388fa917fb",  
  "address": "70 Summer walk, USA",  
  "key": "qaclick123"  
}
```

### Sample Response for the Provided Place\_Id

```
{  
  "location": {  
    "lat": -38.383494,  
    "lng": 33.427362  
  },  
  "accuracy": 50,  
  "name": "Frontline house",  
  "phone_number": "(+91) 983 893 3937",  
  "address": "29, side layout, cohen 09",  
  "types": ["shoe park", "shop"],  
  "website": "http://google.com",  
  "language": "French-IN"  
}
```

---

### Library API :

BaseURI : http://216.10.245.166

1. **Resource** : Library/Addbook.php      **Method** : POST

### Input Payload : Json:

```
{
```

```
"name": "Learn Appium Automation with Java",
"isbn": "bcd",
"aisle": "227",
"author": "John foe"
}
```

### Output Json

```
{
  "Msg": "successfully added",
  "ID": "bcd227"
}
```

1. **Resource** : /Library/GetBook.php?AuthorName=somename      **Method** : GET

### Output Json :

Output the array of Json object books with all below details

```
{
  Name : "bookname" ( String)
  Isbn : "A2fdsf" (String)
  Aisle : 32 (Integer)
}
```

1. **Resource** : Library/GetBook.php?ID=3389      - **Method** : GET

### Output Json :

```
{
  "book_name": "Selenium automation using Java",
  "isbn": "a23hd738",
  "aisle": "1223"
}
```

1. **Resource** : /Library/DeleteBook.php      **Method** : POST

### Input Payload : Json:

```
{
  "ID" : "a23h345122332"
}
```

```
}
```

### **Output Response :**

```
{
```

msg : book is successfully deleted"

```
import static io.restassured.RestAssured.given;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;
import io.restassured.RestAssured;
import io.restassured.path.json.JsonPath;
import io.restassured.response.Response;
import files.ReusableMethods;
import files.payLoad;
public class DynamicJson {
```

```
@Test(dataProvider="BooksData")
public void addBook(String isbn,String aisle)
```

```
{
```

```
RestAssured.baseURI="http://216.10.245.166";
Response resp=given().
header("Content-Type","application/json").
body(payLoad.Addbook(isbn,aisle)).
when().
post("/Library/Addbook.php").
then().assertThat().statusCode(200).
extract().response();
JsonPath js= ReusableMethods.rawToJson(resp);
String id=js.get("ID");
System.out.println(id);
```

```
//deleteBBook
```

```
}
```

```
@DataProvider(name="BooksData")
```

```
public Object[][] getData()
```

```
{
```

```
//array=collection of elements
```

```
//multidimensional array= collection of arrays
```

```
return new Object[][] {{"ojfwty","9363"},{"cwetee","4253"}, {"okmfet","533"} };
```

```
}
```

---



```

import io.restassured.response.Response;

public class StaticJson {
    @Test
    public void addBook() throws IOException

    {

        RestAssured.baseURI="http://216.10.245.166";
        Response resp=given().
        header("Content-Type","application/json").
        body(GenerateStringFromResource("C:\\Users\\rahul\\Documents\\Addbookdetails.json")).
        when().
        post("/Library/Addbook.php").
        then().assertThat().statusCode(200).
        extract().response();
        JsonPath js= ReusableMethods.rawToJson(resp);
        String id=js.get("ID");
        System.out.println(id);

        //deleteBOok
    }
    public static String GenerateStringFromResource(String path) throws IOException {

        return new String(Files.readAllBytes(Paths.get(path)));

    }

```

---

```

public class JiraTest {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        RestAssured.baseURI="http://localhost:8080";

        //Login Scenario
        SessionFilter session=new SessionFilter();

        String response=given().relaxedHTTPSValidation().header("Content-
        Type","application/json").body("{\r\n" +
        "  \"username\": \"RahulShetty\", \r\n" +
        "  \"password\": \"XXXX11\" \r\n" +
        "}").log().all().filter(session).when().post("/rest/auth/1/session").then().log().all().extract().response(
        ).asString();
    }

```

```
String expectedMessage ="Hi How are you?";
```

```
//Add comment
```

```
String addCommentResponse = given().pathParam("key", "10101").log().all().header("Content-Type","application/json").body("{\r\n" +  
"  \"body\": \"\"+expectedMessage+\"\", \r\n" +  
"  \"visibility\": {\r\n" +  
"    \"type\": \"role\", \r\n" +  
"    \"value\": \"Administrators\" \r\n" +  
"  } \r\n" +  
"}").filter(session).when().post("/rest/api/2/issue/{key}/comment").then().log().all()  
.assertThat().statusCode(201).extract().response().asString();
```

```
JsonPath js=new JsonPath(addCommentResponse);  
String commentId= js.getString("id");
```

```
//Add Attachment
```

```
given().header("X-Atlassian-Token","no-check").filter(session).pathParam("key", "10101")  
.header("Content-Type","multipart/form-data")  
.multiPart("file",new File("jira.txt")).when()  
post("rest/api/2/issue/{key}/attachments").then().log().all().assertThat().statusCode(200);
```

```
//Get Issue
```

```
String issueDetails=given().filter(session).pathParam("key", "10101")  
.queryParams("fields", "comment")  
.log().all().when().get("/rest/api/2/issue/{key}").then()  
.log().all().extract().response().asString();
```

```
System.out.println(issueDetails);
```

```
JsonPath js1 =new JsonPath(issueDetails);  
int commentsCount=js1.getInt("fields.comment.comments.size()");  
for(int i=0;i<commentsCount;i++)  
{  
String commentIdIssue =js1.get("fields.comment.comments["+i+"].id").toString();  
  
if (commentIdIssue.equalsIgnoreCase(commentId))  
{  
String message= js1.get("fields.comment.comments["+i+"].body").toString();  
System.out.println(message);  
Assert.assertEquals(message, expectedMessage);  
}  
}
```

---

## OAuth 2.0 Contract Details:

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-p0m7ent2hk7suguv4vq22hjcfc43pj.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

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

```
import io.restassured.parsing.Parser;
import io.restassured.path.json.JsonPath;
import io.restassured.response.Response;
import io.restassured.response.ResponseBody;
import pojo.Api;
import pojo.GetCourse;
```

```
public class test1 {
```

```
    public static void main(String[] args) throws InterruptedException {
```

```
        // TODO Auto-generated method stub
```

```
        String url
```

```
        ="https://rahulshettyacademy.com/getCourse.php?state=verifyfdss&code=4%2FvAHBQUZU6o4WJ719NrGBzSELBFVBI9XbxvOtYpmYpeV47bFVExkaxWaF_XR14PHtTZf7ILSEeamywjKwo_BYs9M&scope=email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+openid&aut
```

```
huser=0&session_state=0c32992f0d47e93d273922018ade42d1072b9d1f..a35c&prompt=none#";
```

```
String partialcode=url.split("code=")[1];
String code=partialcode.split("&scope")[0];
System.out.println(code);
String response =
    given()
        .urlEncodingEnabled(false)
        .queryParams("code",code)

        .queryParams("client_id", "692183103107-
p0m7ent2hk7suguv4vq22hjcfhcr43pj.apps.googleusercontent.com")
        .queryParams("client_secret", "erZOWM9g3UtwNRj340YYaK_W")
        .queryParams("grant_type", "authorization_code")
        .queryParams("state", "verifyfjdss")
        .queryParams("session_state",
"ff4a89d1f7011eb34eef8cf02ce4353316d9744b..7eb8")
        // .queryParams("scope",
"email+openid+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email")

        .queryParams("redirect_uri", "https://rahulshettyacademy.com/getCourse.php")
        .when().log().all()
        .post("https://www.googleapis.com/oauth2/v4/token").asString();
// System.out.println(response);
JsonPath jsonPath = new JsonPath(response);
String accessToken = jsonPath.getString("access_token");
System.out.println(accessToken);
String r2= given().contentType("application/json").
queryParams("access_token", accessToken).expect().defaultParser(Parser.JSON)
.when()
    .get("https://rahulshettyacademy.com/getCourse.php")
.asString();
System.out.println(r2);

}
```

---

## Request and Response Spec Builders :

ADD PLACE :

```
RestAssured.baseURI="XXXX";
```

```
Response res=given().queryParams("key", "qaclick123").header("Content-Type","application/json")
```

```
.body(add_place_json)
.when().post("/maps/api/place/add/json").
then().assertThat().statusCode(200).contentType("application/json")
extract().response();
```

GET\_PLACE

```
RestAssured.baseURI="XXXX";
```

```
Response res=given().queryParams("key", "qaclick123").header("Content-Type","application/json")
when (). get("/maps/api/place/get/json").
then().assertThat().statusCode(200).contentType("application/json").extract().response();
```

DELETE\_PLACE

```
RestAssured.baseURI="XXXX";
```

```
Response res=given().queryParams("key", "qaclick123").header("Content-Type","application/json")
.body("delete_place_json")
.when().post("/maps/api/place/delete/json").
then().assertThat().statusCode(200).contentType("application/json").extract().response();
```

Build -Request Spec Builder-

```
req= new RequestSpecBuilder().setContentType(ContentType.JSON)

        .setBaseUri("XXXX")
        .addQueryParam("key", "qaclick123")
        .build();
```

```
given().spec (req ).body(add_place_json) .post("/maps/api/place/add/json).
```

Build Response Spec Builder:

```
res = new ResponseSpecBuilder().expectStatusCode(200).expectContentType(ContentType.JSON).
build();
```

```
then().spec(re).extract().response();
```

Rewriting Test with Request and Response Spec Builder :

```
given().spec(req).body(add_place_json).post("/maps/api/place/add/json).
```

```
then().spec(res).extract().response();
```

