

Rest Assured API Basics

Application Programming Interface(API)

is an interface or communication protocol between a client and a server intended to simplify the building of client-side software. It has been described as a 'contract' between the client and the server, such that if the client makes a request in specific format, it will always get a response in a specific format or initiate a defined action.

End point/ Base URL: 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

CRUD operations:

C - Create

R - Retrieve

U - Update

D - Delete

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

In this example -> maps, search, images are the resources developed by API

developers

In this example -> [google.com](https://www.google.com) is the endpoint/ base URL where these above resources/ API resides

Every API should have resource & base URL - these are mandatory because without these details we can send any request to API server

There are 2 types of parameters when calling API to API server:

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

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

2. 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

Postman

ADD-DeletePlaceAPIs.docx

Word Document · 12 KB



Google+Place+APIs.postman_collection.json

JSON · 3 KB



POST -> is a superset of DELETE, PUT requests.
POST can be used for altering the database as some companies only this method for every CRUD operation

RestAssured

Rest Assured is a java library that is used to test RESTful Web Services

3 simple steps to setup Rest Assured project

- Install Java and set in System Variables
- Install Eclipse and Create Java Maven project
- Configure Rest Assured Jars into project

> Command to check java version installed in Mac

```
/usr/libexec/java_home -V  
java -version
```

>Command to update the path to Mac shell(to set environment variables in Mac)

```
Zshrc  
vi ~/.zshrc  
source ~/.zshrc  
Esc :wq  
echo $JAVA_HOME  
printenv
```

>To match JSON response to actual response we need Hamcrest maven dependency in API automation

In RestAssured package, we have libraries or methods which are static in nature

```
Import static io.restassured.RestAssured.*;
```

Below methods belongs to rest assured static package:

- given()
- when()
- then()

In hamcrest static package we have the equalTo() method
 import static org.hamcrest.Matchers.*;

JsonPath:

JsonPath -> is a class which takes 'string' as input and convert into 'json' and it will help to 'parse the json'.
 String response= \$jsonResponseBody.response().asString();
 JsonPath js = new JsonPath(response);

JsonOnlineEditor: <https://jsoneditoronline.org/>

Parsing the Complex Nested Json

Mock response data as we do not have API ready to test for below mock response

```
String response = "
{
  "dashboard": {
    "purchasedAmount": 910,
    "website": "rahulshettyacademy.com"
  },
  "courses": [
    {
      "title": "Selenium Python",
      "price": 50,
      "copies": 6
    },
    {
      "title": "Cypress",
      "price": 40,
      "copies": 4
    },
    {
      "title": "RPA",
      "price": 45,
      "copies": 10
    }
  ]
}
";
JsonPath js = new JsonPath(response);
```

1. Print No of courses returned by API

2. Print Purchase Amount
3. Print Title of the first course
4. Print All course titles and their respective Prices
5. Print no of copies sold by RPA Course
6. Verify if Sum of all Course prices matches with Purchase Amount

Advanced Payload Creation Strategies:

1. Dynamically build json payload with external data inputs
2. Parameterize the API Tests with multiple data sets
3. How to send static Json files(payload) directly into Post method of Rest Assured
4. Feed Json payload Using Excel using HashMap
5. POJO classes to build payload

Ex: Library API ->

Addbook

120

Bookname, author name, isbn

Library+API.docx

Word Document · 16 KB



Data Parameterization using @DataProvider

array = collection of elements

multidimensional = collection of arrays

```
@DataProvider(name = "BooksData")
public Object[][] getData() {
    Object[][] object = new Object[][] {{"ajsaw", "1212"},
{"ajsw", "1222"}, {"ajqaw", "1112"}};
    return object;
}
```

Postman collections:

Library+API.postman_collection.json

JSON · 2 KB



Library API created by me.postman_collection.json

JSON · 546 KB



Google+Place+APIs.postman_collection.json

JSON · 3 KB



RSA_Maps.postman_collection.json

JSON · 21 KB



Documents:

Library+API.docx

Word Document · 16 KB



ADD-DeletePlaceAPIs.docx

Word Document · 12 KB



courses.pages

Pages Document · 107 KB

