API TESTING THROUGH BROWSER USING CODE SNIPPETS

What is API

API (**A**pplication **P**rogramming **I**nterface) enables communication and data exchange between two separate software systems. A software system implementing an API contains functions/sub-routines which can be executed by another software system.

What is API Testing

API TESTING is a software testing type that validates Application Programming Interfaces (APIs). The purpose of API Testing is to check the functionality, reliability, performance, and security of the programming interfaces. In API Testing, instead of using standard user inputs(keyboard) and outputs, we use software to send calls to the API, get output, and note down the system's response. API tests are very different from GUI Tests and won't concentrate on the look and feel of an application. It mainly concentrates on the business logic layer of the software architecture.

The API testing can by performed by various tools like POSTMAN, SoapUI and Reset-Assured but this document through coding using python.

What is REST API

REST suggests to create an object of the data requested by the client and send the values of the object in response to the user. For example, if the user is requesting for a movie in Bangalore at a certain place and time, then one can create an object on the server-side. So, over here, we have an object and we are sending the state of an object. This is why REST is known as Representational State Transfer.

What is HTTP

The Hypertext Transfer Protocol (HTTP) is designed to enable communications between clients and servers. HTTP works as a request-response protocol between a client and server. A web browser may be the client, and an application on a computer that hosts a website may be the server.

Example: A client (browser) submits an HTTP request to the server; then the server returns a response to the client. The response contains status information about the request and may also contain the requested content.

Most common HTTP methods:

1. **GET** : The GET method is used to retrieve information from the given server using a given URI. Requests using GET should only retrieve data and should have no other effect on the data.

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

3. PUT : PUT is used to send data to a server to create/update a resource. Replaces all the current representations of the target resource with the uploaded content.

4. PATCH : PATCH is used to update partial resources. For instance, when you only need to update one field of the resource, putting a complete resource representation might be cumbersome and utilizes more bandwidth.

5. HEAD : HEAD is almost identical to GET, but without the response body. HEAD transfers the status line and the header section only.

6. DELETE : The DELETE method deletes the specified resource.

7. OPTIONS : The OPTIONS method describes the communication options for the target resource.

Testing GET Requests

Suppose there is an API which fetches the user information of a particular application. To test this we will have to use GET request. The GET request is explained below:

For sample requests, visit https://reqres.in/

From this site we will select a rest API(in this testing the example used) **" //api/users?page=2"**

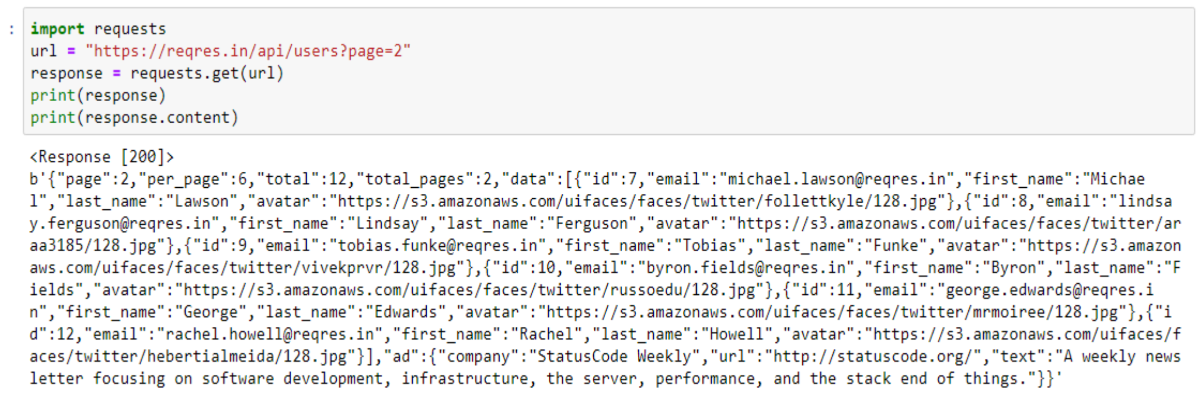
For doing this we will install request library in python. The Requests package isn't part of Python's standard library. But the way that it wraps up Python's standard HTTP functionality into a simple, elegant interface makes it one of the most widely used external libraries.

After this we have to install Jason. JSON Server is a simple project that helps us to setup a REST API with CRUD operations very fast.

To install jsaon :- **pip install Jason**

When one makes a request to a URI, it returns a response. This Response object in terms of python is returned by requests.method(), method being – get, post, put, etc. Response is a powerful object with lots of functions and attributes that assist in normalizing data or creating ideal portions of code.

The response here displays the content of the respective API.



```python
import requests
url = "https://reqres.in/api/users?page=2"
response = requests.get(url)
print(response)
print(response.content)
```

```
<Response [200]>
b'{"page":2,"per_page":6,"total":12,"total_pages":2,"data":[{"id":7,"email":"michael.lawson@reqres.in","first_name":"Michae
l","last_name":"Lawson","avatar":"https://s3.amazonaws.com/uifaces/faces/twitter/follettkyle/128.jpg"},{"id":8,"email":"lindsa
y.ferguson@reqres.in","first_name":"Lindsay","last_name":"Ferguson","avatar":"https://s3.amazonaws.com/uifaces/faces/twitter/ar
aa3185/128.jpg"},{"id":9,"email":"tobias.funke@reqres.in","first_name":"Tobias","last_name":"Funke","avatar":"https://s3.amazon
aws.com/uifaces/faces/twitter/vivekprvr/128.jpg"},{"id":10,"email":"byron.fields@reqres.in","first_name":"Byron","last_name":"F
ields","avatar":"https://s3.amazonaws.com/uifaces/faces/twitter/russoedu/128.jpg"},{"id":11,"email":"george.edwards@reqres.i
n","first_name":"George","last_name":"Edwards","avatar":"https://s3.amazonaws.com/uifaces/faces/twitter/mrmoiree/128.jpg"},{"i
d":12,"email":"rachel.howell@reqres.in","first_name":"Rachel","last_name":"Howell","avatar":"https://s3.amazonaws.com/uifaces/f
aces/twitter/hebertialmeida/128.jpg"}],"ad":{"company":"StatusCode Weekly","url":"http://statuscode.org/","text":"A weekly news
letter focusing on software development, infrastructure, the server, performance, and the stack end of things."}}'
```

Fig: Response received

References

- https://www.mulesoft.com/resources/api/what-is-an-api
- https://developer.basespace.illumina.com/docs/content/documentation/rest-api/api-using-a-browser
- https://www.geeksforgeeks.org/