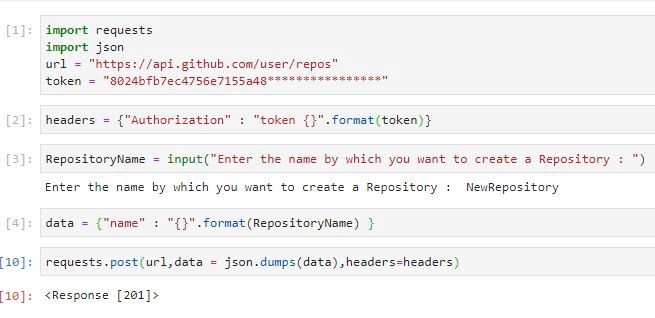
We will be using Python to make the API calls .

**1. Create a Repository**



Response [201] implies that repo has been created

Replace the token with your own generated token in line 1 to create a repository in your account. After executing the above code with a response of 201 go and check out your account for the newly created repository. If you find it congratulate yourself, you made your first call🥰👍.

Any data that is needed to be sent along with the request needs to be in JSON format. The token is to be included inside headers.

json.dumps() function transforms the json file(or dictionary in python) to a json string.

[Click Here](https://developer.github.com/v3/repos/#create-a-repository-for-the-authenticated-user) to view documentation.

**2. Delete a Repository**



Response 204 implies successful deletion

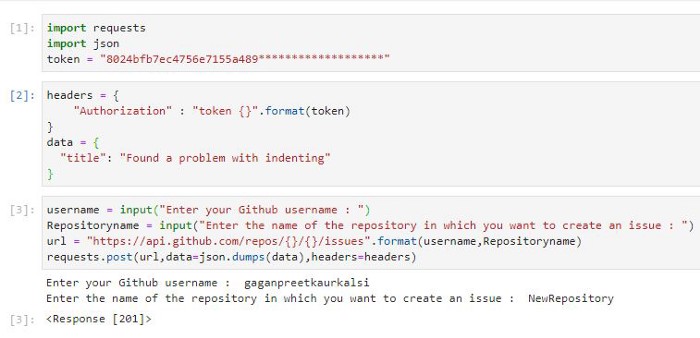
Here we make a delete request as we want to make a deletion and the process requires Authorization so we need to pass a token. The process of construction of url is the same as I explained before.

[Click Here](https://developer.github.com/v3/repos/#delete-a-repository) for documentation.



:owner means you need to fill in your username. Same with others.

**3. Create an Issue**



Response 201 — success

Here data is a JSON containing ‘Parameters’ as I explained before. Parameters are the specifications.

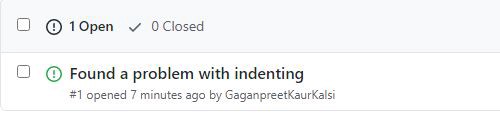
[Click Here](https://developer.github.com/v3/issues/#create-an-issue) for documentation.

**4. Comment on an Issue**



Response 201 — success

Issue number can be found out under Issues tab in your repository prefixed with a hashtag #.



Issue number 1 (#1)

**5. Close an Issue**

This was particularly the one which caused me a lot of trouble. I couldn’t find any documentation for it on web. But after analyzing the JSON response I found the solution. You can print the JSON reponse file by writing following 2 lines :-

output = requests.post(url,data=json.dumps(data),headers=headers )

output.json() #*This line will print the response in JSON format*



This is a part of JSON response received while creating an Issue

To close an issue change the state to close and send it with post request.



Response 200 — Success



Issue closed

**6. Create Pull Request**

For creating a pull request ->

* Create a branch — *provide master with a friend to communicat*e
* Create a file in the newly created branch which can be pulled into master.

The setup is ready. Now you just need to run the code.



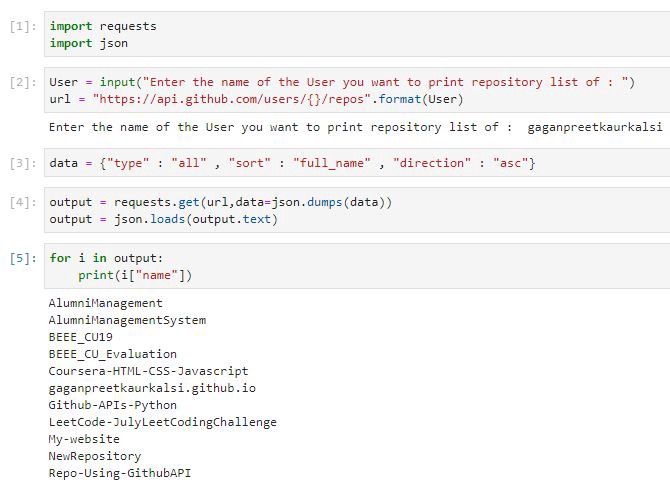
Response 201 — success

**Parameters** — I guess title and body must be clear from the name itself.

The file needs to be pulled from head branch(Newly created branch) to base branch(Master branch). I created a branch named pull-request.

Now what you need to focus on is the Accept key/object in headers. Without this the request will not be made and you will be thrown an error. While making some API requests you need to mention it along with the token inside headers. You will find it mentioned in the [documentation](https://developer.github.com/v3/pulls/#create-a-pull-request).

**7. List Repositories for a user**



How to access and use APIs like GITHUB API using python’s JSON and request library.

# importing the required modules

import json

import requests

# prompting the user to input the required data

username = input("Please Enter The Username: ")

repo = input("Please Enter The Repository Name: ")

# building the url for API

url = f'https://api.github.com/repos/{username}/{repo}'

# sending an HTTP request using request module

r = requests.get(url)

# parsing the json and converting it into python dictionary

data = json.loads(r.text)

# displaying the repository data

print("\n")

print('Repository Name:',data['name'])

print('Repository Description:',data['description'])

print("Programming Language Used:", data['language'])

print("Repository Created:", data['created\_at'])

print("Repository Size:", data['size']/1024, "Mb")

print("Total stars:", data['stargazers\_count'])

print("Total forks:", data['forks\_count'])

print("Total watchers:", data['watchers\_count'])

print("\n")

# End of the program