	Marwadi University **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Practical based on Interacting with Web APIs	
Experiment No: 21	**Date:**	**Enrollment No: 92510133028**

Aim: Practical based on Interacting with Web APIs

IDE:

To interact with web APIs using Python and libraries like requests. The practical will involve making GET and POST requests, handling responses, and parsing JSON data.

Library Installation

```
pip install requests
```

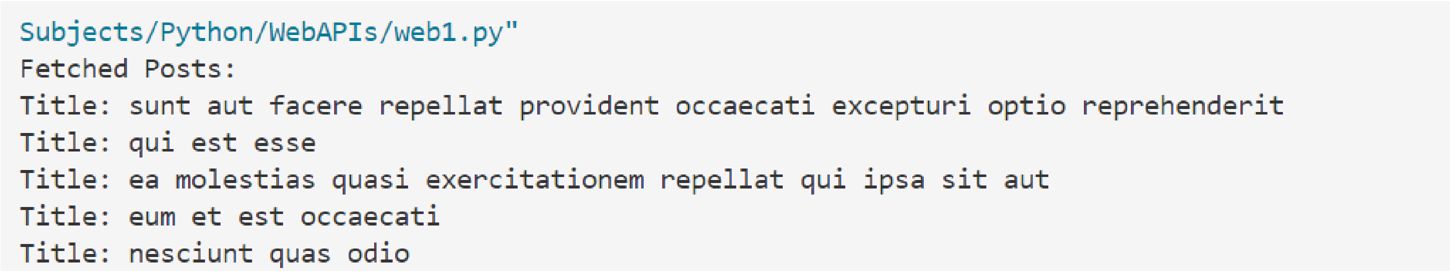
Select an API: Choose a public API for testing. For this tutorial, we'll use the JSONPlaceholder API, a free fake online REST API for testing and prototyping.

Making a GET Request

Fetch a list of posts.

```
import requests


# Define the API endpoint

url = 'https://jsonplaceholder.typicode.com/posts'


# Make a GET request

response = requests.get(url)


# Check the status code

if response.status_code == 200:

    # Parse JSON response

    posts = response.json()
    print("Fetched Posts:")

    for post in posts[:5]:  # Print the first 5 posts

        print(f"Title: {post['title']}")

else:

    print(f"Failed to retrieve posts. Status code: {response.status_code}")
```

Output:



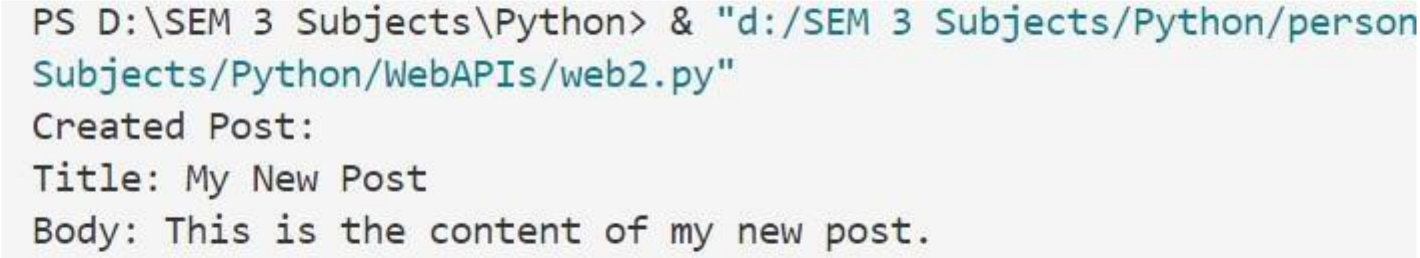
```
Subjects/Python/WebAPIs/web1.py"
Fetched Posts:
Title: sunt aut facere repellat provident occaecati excepturi optio reprehenderit
Title: qui est esse
Title: ea molestias quasi exercitationem repellat qui ipsa sit aut
Title: eum et est occaecati
Title: nesciunt quas odio
```

Making a POST Request

Create a new post.

```
import requests



# Define the API endpoint

url = 'https://jsonplaceholder.typicode.com/posts'



# Define the data to send

new_post = {

    "title": "My New Post",

    "body": "This is the content of my new post.",

    "userId": 1
}


# Make a POST request

response = requests.post(url, json=new_post)


# Check the status code

if response.status_code == 201:

    # Parse JSON response

    created_post = response.json()

    print("Created Post:")

    print(f"Title: {created_post['title']}")

    print(f"Body: {created_post['body']}")

else:

    print(f"Failed to create post. Status code: {response.status_code}")
```

Output:



```
PS D:\SEM 3 Subjects\Python> & "d:/SEM 3 Subjects/Python/person
Subjects/Python/WebAPIs/web2.py"
Created Post:
Title: My New Post
Body: This is the content of my new post.
```

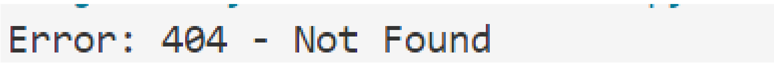
Error Handling

Handle potential errors in API requests.

```
import requests
```

```
url = 'https://jsonplaceholder.typicode.com/posts/1000'  # Non-existent post


response = requests.get(url)


if response.status_code == 200:
    post = response.json()
    print("Fetched Post:", post)
else:
    print(f"Error: {response.status_code} - {response.reason}")
```

Output:



```
Error: 404 - Not Found
```

Post Lab:

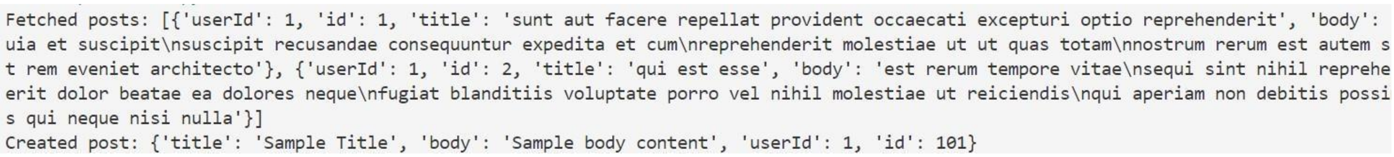
https://github.com/keshvi1234/PWP_experiment

Building a Simple API Client (Write a simple Python class that encapsulates the functionality for making GET and POST requests to the JSONPlaceholder API. Include methods for fetching posts and creating a new post.)

Code:

```
import requests

class JSONPlaceholderClient:
    def_init_(self):
        self.base_url = "https://jsonplaceholder.typicode.com"

    def fetch_posts(self):
        """Fetches a list of posts."""
        try:
            response = requests.get(f"{self.base_url}/posts")
            response.raise_for_status()
            return response.json()
        except requests.exceptions.RequestException as e:
            print("Error fetching posts:", e)
            return None

    def create_post(self, title, body, user_id):
        """Creates a new post."""
        payload = {"title": title, "body": body, "userId": user_id}
        try:
            response = requests.post(f"{self.base_url}/posts", json=payload)
            response.raise_for_status()
            return response.json()
        except requests.exceptions.RequestException as e:
            print("Error creating post:", e)
            return None

if___name___== "__main__":
    client = JSONPlaceholderClient()
    posts = client.fetch_posts()
    if posts:
        print("Fetched posts:", posts[:2])
        new_post = client.create_post("Sample Title", "Sample body content", 1)
        if new_post:
            print("Created post:", new_post)
```

Output:



```
Fetched posts: [{'userId': 1, 'id': 1, 'title': 'sunt aut facere repellat provident occaecati excepturi optio reprehenderit', 'body':
uia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem s
t rem eveniet architecto'}, {'userId': 1, 'id': 2, 'title': 'qui est esse', 'body': 'est rerum tempore vitae\nsequi sint nihil reprehe
erit dolor beatae ea dolores neque\nfugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis\nqui aperiam non debitis possi
s qui neque nisi nulla'}]
Created post: {'title': 'Sample Title', 'body': 'Sample body content', 'userId': 1, 'id': 101}
```