

 Marwadi University <small>Marwadi Chandarana Group</small>	 NAAC A+	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:08-11-2025	Enrollment No:92400133131

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()
```

 Marwadi University <small>Marwadi Chandarana Group</small>	 NAAC A+	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: 08-11-2025	Enrollment No: 92400133131

```

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:-

```

```

SyntaxError: invalid syntax
PS D:\python\vs code> & C:/Users/ACER/AppData/Local/Programs/Python/Python314/python.
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
PS D:\python\vs code>

```

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.",
```

 Marwadi University <small>Marwadi Chandarana Group</small>	 NAAC A+	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:08-11-2025	Enrollment No:92400133131

```
"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:-
```

```
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
```



Subject: Programming With Python (01CT1309)

Aim: Practical based on Interacting with Web APIs

Experiment No: 21

Date:08-11-2025

Enrollment No:92400133131

```
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:

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 SimpleAPIClient:
```

```
    def __init__(self):
```

```
        self.base_url = "https://jsonplaceholder.typicode.com"
```

 Marwadi University <small>Marwadi Chandarana Group</small>	 NAAC A+	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: 08-11-2025	Enrollment No: 92400133131

```
def get_posts(self):
    response = requests.get(f"{self.base_url}/posts")
    return response.json()
```

```
def create_post(self, title, body, user_id):
    data = {"title": title, "body": body, "userId": user_id}
    response = requests.post(f"{self.base_url}/posts", json=data)
    return response.json()
```

```
client = SimpleAPIClient()
print(client.get_posts()[:2])
print(client.create_post("Hello", "This is a test post", 1))
```

Output :-

```
PS D:\paython\vs code> & C:/Users/ACER/AppData/Local/Programs/Python/Python314/python.exe "d:/paython/vs code/post_lab21.py"
[{"userId": 1, "id": 1, "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit", "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"}, {"userId": 1, "id": 2, "title": "qui est esse", "body": "est rerum tempore vitae\nsequi sint nihil reprehenderit dolor beatae ea dolores neque\nfugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis\nnisi aperiam non debitis possimus qui neque nisi nulla"}]
{'title': 'Hello', 'body': 'This is a test post', 'userId': 1, 'id': 101}
```

Github link :- :

<https://github.com/farhanweb404/farhankaladiya.git>