

 Marwadi University <small>Marwadi Chandrana 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:	Enrollment No: 92400133181

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 Chandrana 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:	Enrollment No: 92400133181

```

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}")

```

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)
```

 Marwadi University Marwadi Chandrana Group	NAAC  A+	
Subject: Programming With Python (01CT1309)	Aim: Practical based on Interacting with Web APIs	
Experiment No: 21	Date:	Enrollment No: 92400133181

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}")
```

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}")
```



Subject: Programming With Python (01CT1309)

Aim: Practical based on Interacting with Web APIs

Experiment No: 21

Date:

Enrollment No: 92400133181

```
1 import requests
2
3 url = 'https://jsonplaceholder.typicode.com/posts'
4
5 response = requests.get(url)
6
7 if response.status_code == 200:
8     posts = response.json()
9     print("Fetched Posts:")
10    for post in posts[:5]:
11        print(f"Title: {post['title']}")
12 else:
13     print(f"Failed to retrieve posts. Status code: {response.status_code}")
14
15 import requests
16
17 # Define the API endpoint
18 url = 'https://jsonplaceholder.typicode.com/posts'
19
20 # Define the data to send
21 new_post = {
22     "title": "My New Post",
23     "body": "This is the content of my new post.",
24     "userId": 1
25 }
26
27 # Make a POST request
28 response = requests.post(url, json=new_post)
29
```



Subject: Programming With Python (01CT1309)

Aim: Practical based on Interacting with Web APIs

Experiment No: 21

Date:

Enrollment No: 92400133181

```
29
30     # Check the status code
31     if response.status_code == 201:
32         # Parse JSON response
33         created_post = response.json()
34         print("Created Post:")
35         print(f"Title: {created_post['title']}")
36         print(f"Body: {created_post['body']}")
37     else:
38         print(f"Failed to create post. Status code: {response.status_code}")
39 import requests
40
41 url = 'https://jsonplaceholder.typicode.com/posts/1000'
42
43 response = requests.get(url)
44
45 if response.status_code == 200:
46     post = response.json()
47     print("Fetched Post:", post)
48 else:
49     print(f"Error: {response.status_code} - {response.reason}")
50
```

Output:



Subject: Programming With Python (01CT1309)

Aim: Practical based on Interacting with Web APIs

Experiment No: 21

Date:

Enrollment No: 92400133181

```
...: 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}")
...:
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
Created Post:
Title: My New Post
Body: This is the content of my new post.
Error: 404 - Not Found
```

In [21]:

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:



Subject: Programming With Python (01CT1309)

Aim: Practical based on Interacting with Web APIs

Experiment No: 21

Date:

Enrollment No: 92400133181

```
1 import requests
2
3 class JSONPlaceholderClient:
4     BASE_URL = "https://jsonplaceholder.typicode.com"
5
6     def __init__(self):
7         self.session = requests.Session()
8
9     def get_posts(self):
10        """Fetch all posts"""
11        url = f"{self.BASE_URL}/posts"
12        response = self.session.get(url)
13        return response.json()
14
15    def get_post(self, post_id):
16        """Fetch a single post by ID"""
17        url = f"{self.BASE_URL}/posts/{post_id}"
18        response = self.session.get(url)
19        return response.json()
20
21    def create_post(self, title, body, user_id):
22        """Create a new post"""
23        url = f"{self.BASE_URL}/posts"
24        payload = {
25            "title": title,
26            "body": body,
27            "userId": user_id
28        }
29        response = self.session.post(url, json=payload)
30        return response.json()
31
```



Subject: Programming With Python (01CT1309)

Aim: Practical based on Interacting with Web APIs

Experiment No: 21

Date:

Enrollment No: 92400133181

```
31
32     # Example usage:
33     if __name__ == "__main__":
34         client = JSONPlaceholderClient()
35
36         print("Fetching posts...")
37         posts = client.get_posts()
38         print(posts[:2])  # print first 2 posts
39
40         print("\nFetching post 1...")
41         print(client.get_post(1))
42
43         print("\nCreating a new post...")
44         result = client.create_post("My Title", "This is the body", 1)
45         print(result)
46
```

Output:

```
In [21]: %runfile C:/Users/devah/Documents/PWP/untitled4.py --wdir
Fetching posts...
[{'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\nrefugiat blanditiis voluptate porro vel nihil
molestiae ut reiciendis\nqui aperiam non debitis possimus qui neque nisi
nulla'}]

Fetching post 1...
{'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'}

Creating a new post...
{'title': 'My Title', 'body': 'This is the body', 'userId': 1, 'id': 101}
```

GitHub:

<https://github.com/mallaadisrinivasu132035-code/python.git>