	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: 92400133069**

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}")
```

Making a POST Request

Create a new post.

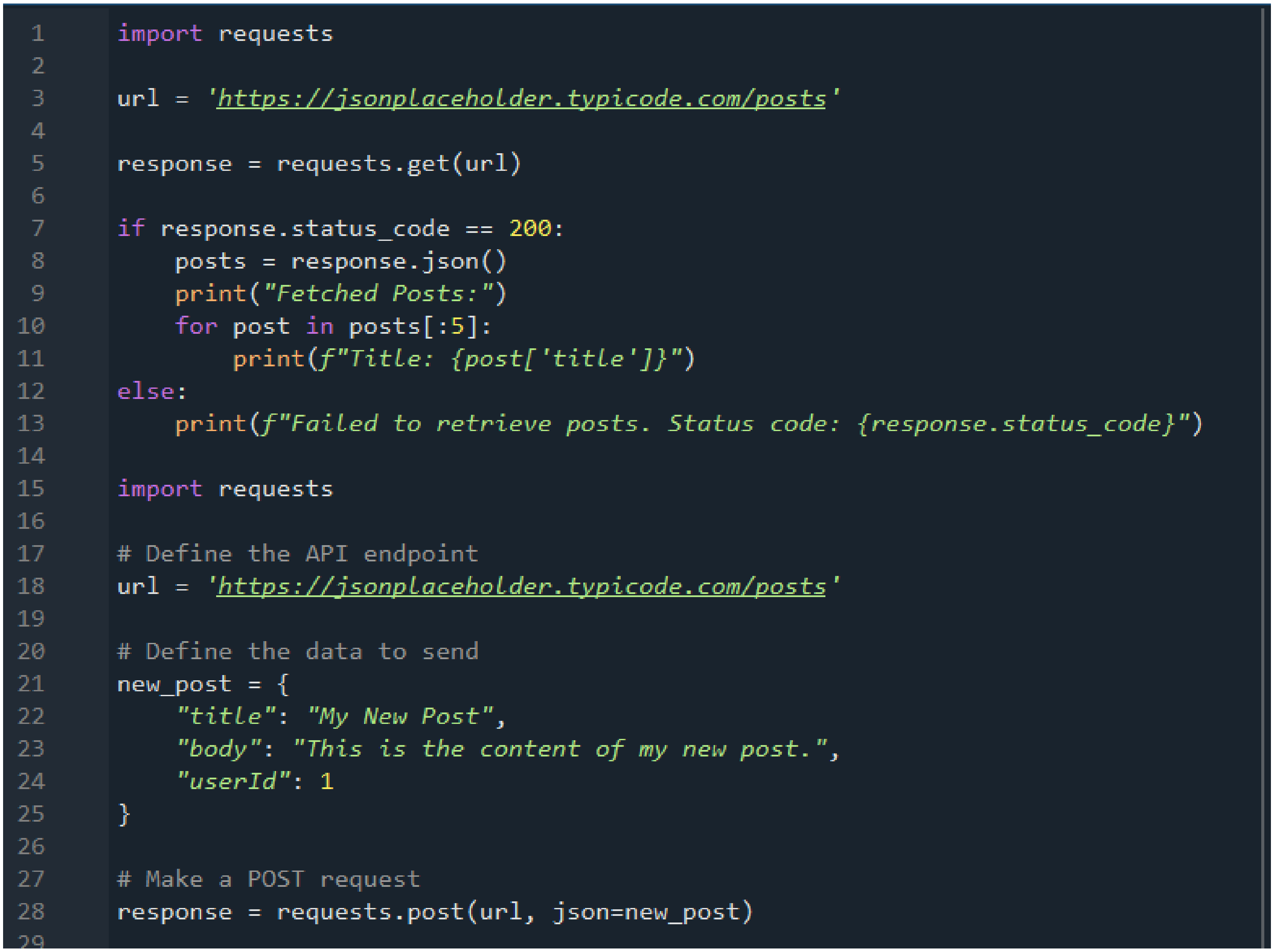
```
import requests


# 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}")
```

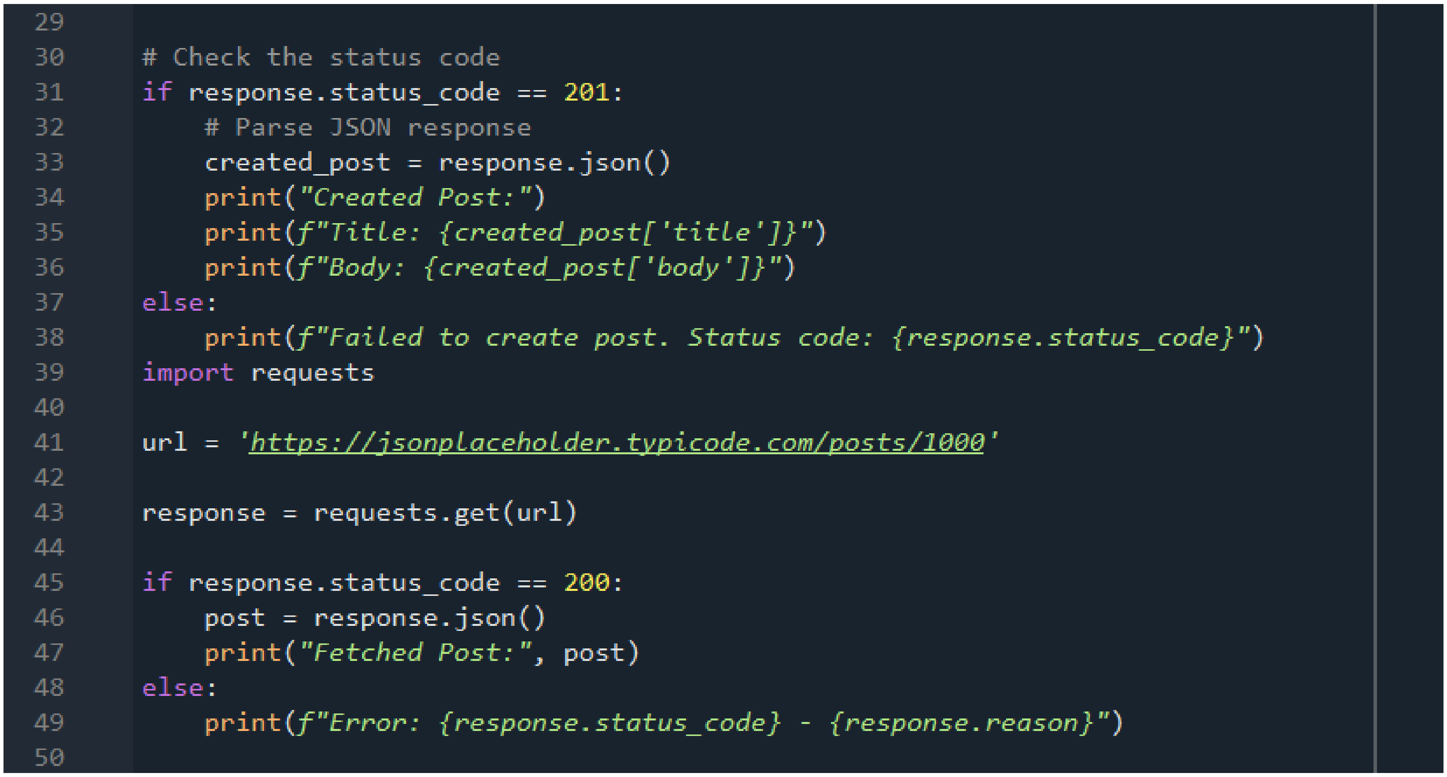
Error Handling

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}")
```

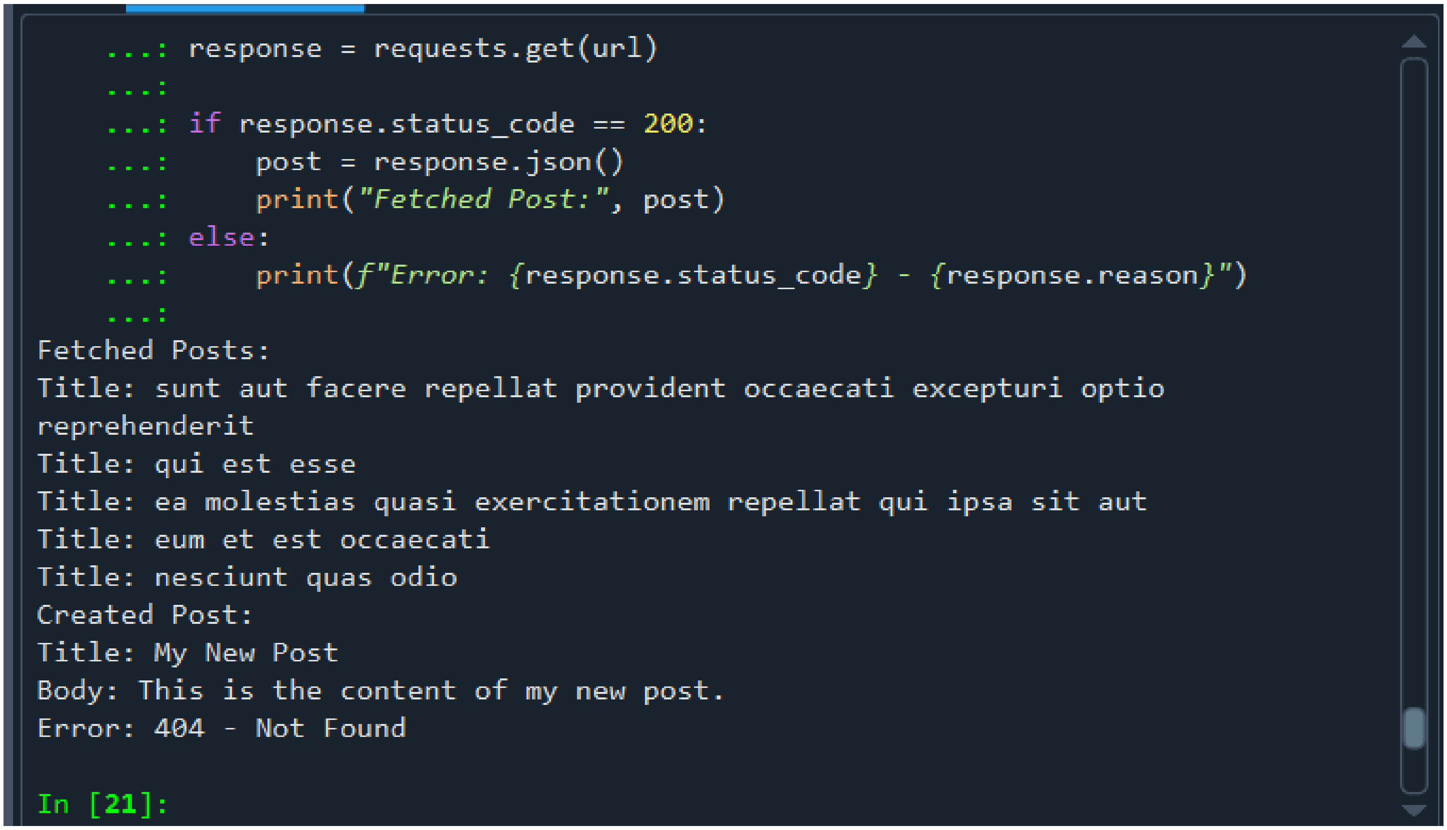


```python
import requests

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

response = requests.get(url)

if response.status_code == 200:
    posts = response.json()
    print("Fetched Posts:")
    for post in posts[:5]:
        print(f"Title: {post['title']}")
else:
    print(f"Failed to retrieve posts. Status code: {response.status_code}")

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}")
import requests

url = 'https://jsonplaceholder.typicode.com/posts/1000'

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:

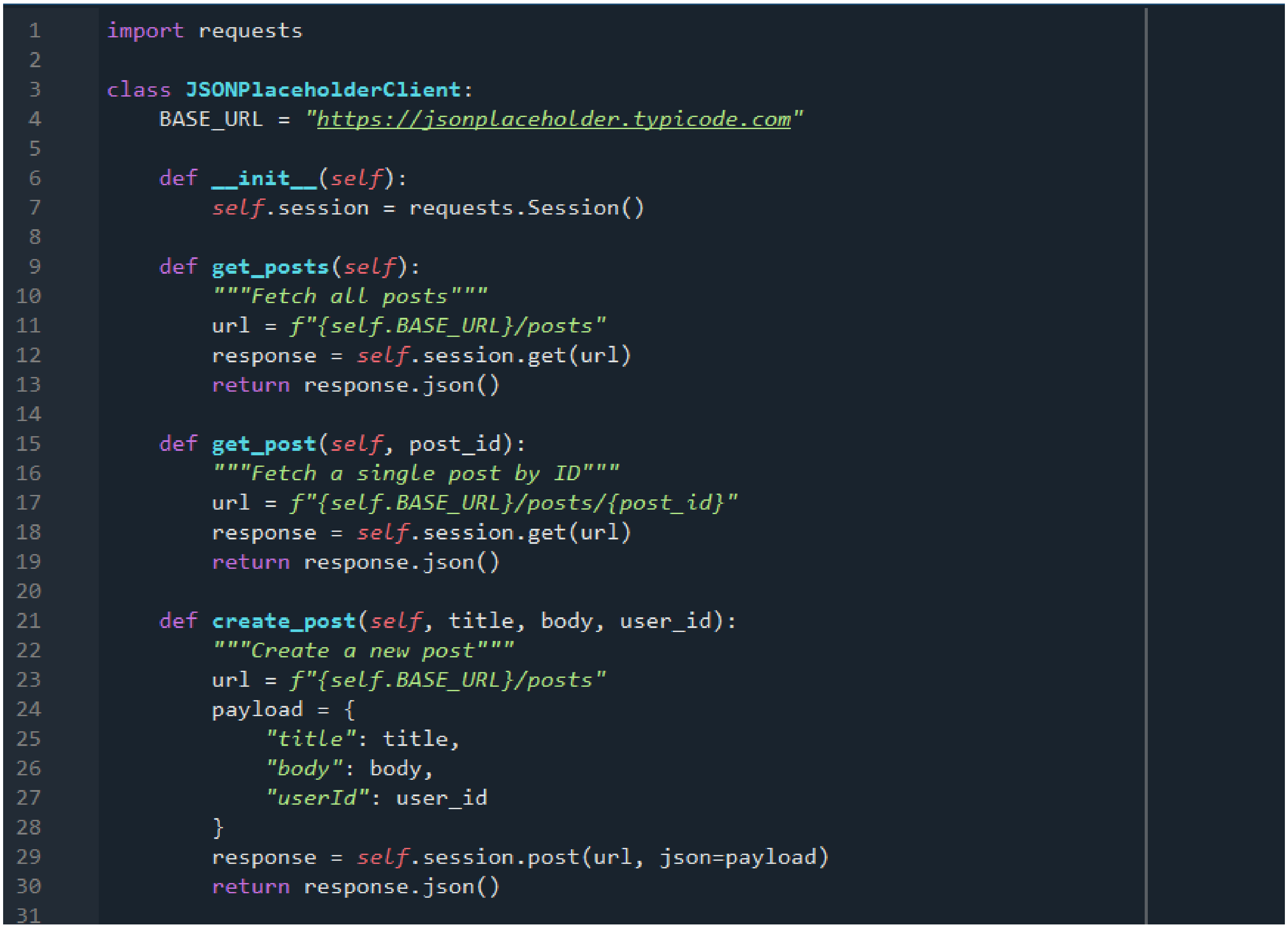


```
    ...: response = requests.get(url)
    ...:
    ...: if response.status_code == 200:
    ...:     post = response.json()
    ...:     print("Fetched Post:", post)
    ...: else:
    ...:     print(f"Error: {response.status_code} - {response.reason}")
    ...:
Fetched Posts:
Title: sunt aut facere 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]:
```

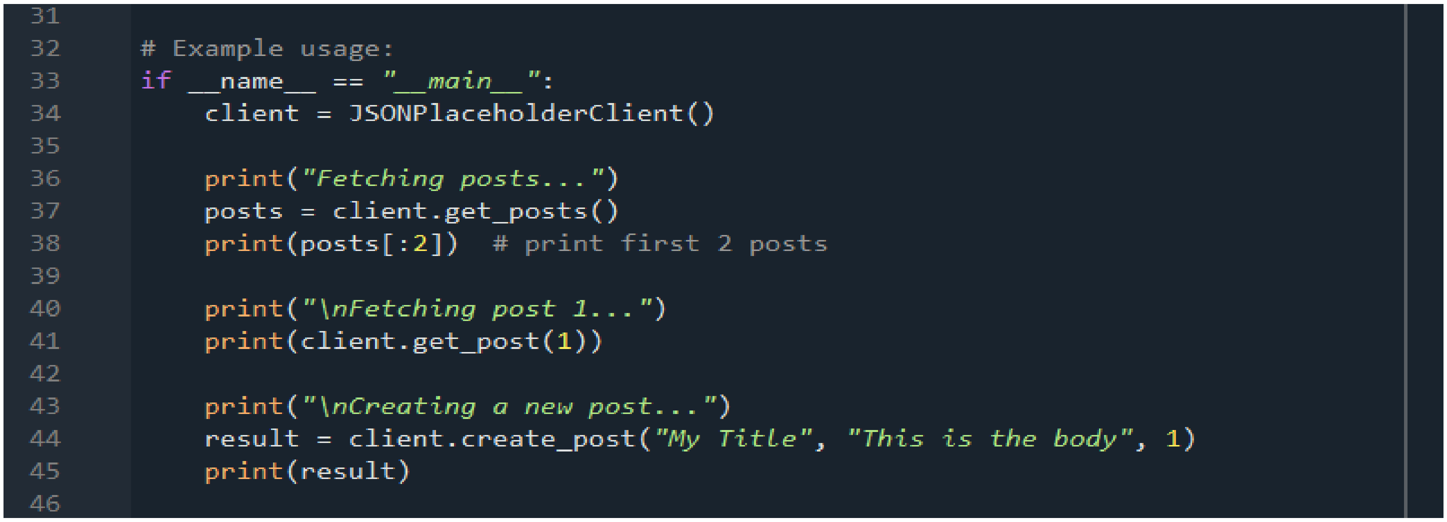
Post Lab:

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:

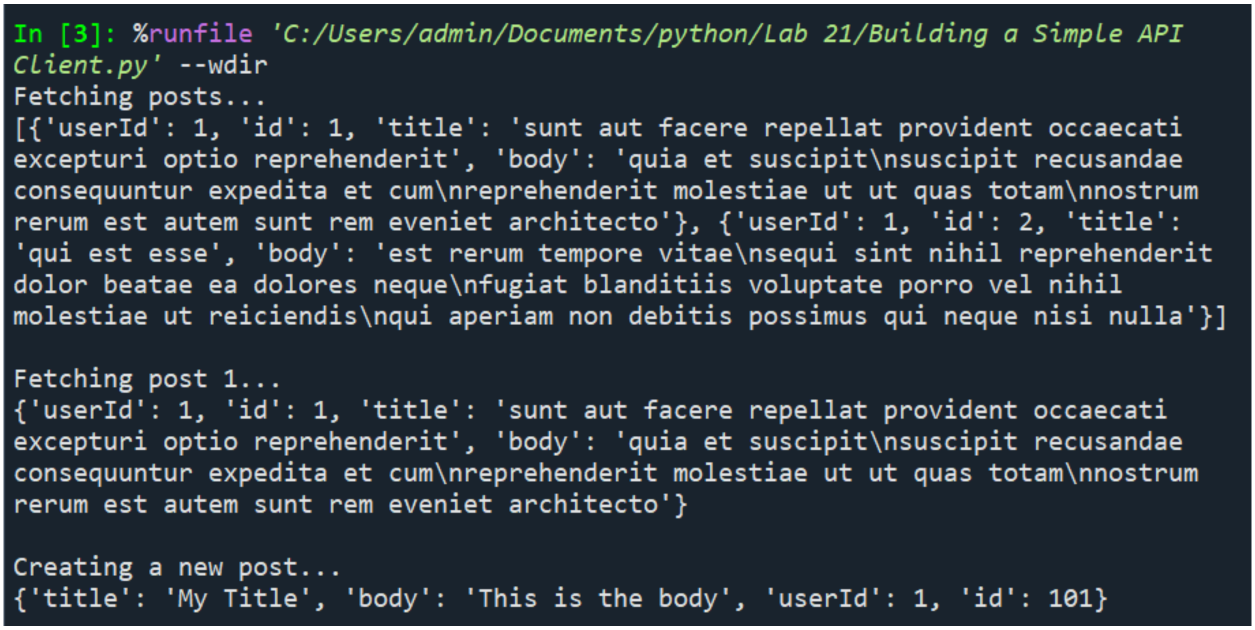


```python
import requests

class JSONPlaceholderClient:
    BASE_URL = "https://jsonplaceholder.typicode.com"

    def __init__(self):
        self.session = requests.Session()

    def get_posts(self):
        """Fetch all posts"""
        url = f"{self.BASE_URL}/posts"
        response = self.session.get(url)
        return response.json()

    def get_post(self, post_id):
        """Fetch a single post by ID"""
        url = f"{self.BASE_URL}/posts/{post_id}"
        response = self.session.get(url)
        return response.json()

    def create_post(self, title, body, user_id):
        """Create a new post"""
        url = f"{self.BASE_URL}/posts"
        payload = {
            "title": title,
            "body": body,
            "userId": user_id
        }
        response = self.session.post(url, json=payload)
        return response.json()

```



```python

# Example usage:
if __name__ == "__main__":
    client = JSONPlaceholderClient()

    print("Fetching posts...")
    posts = client.get_posts()
    print(posts[:2])  # print first 2 posts

    print("\nFetching post 1...")
    print(client.get_post(1))

    print("\nCreating a new post...")
    result = client.create_post("My Title", "This is the body", 1)
    print(result)

```

Output:



```
In [3]: %runfile 'C:/Users/admin/Documents/python/Lab 21/Building a Simple API
Client.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\nfugiat 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/Shashi-kumar-g/Python-Programs.git