
 Marwadi University Marwadi Chandarana Group 	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: 92400133055

[GITHUB](#)

Aim: Practical based on Interacting with Web APIs

IDE: Visual Studio Code

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:

```
Fetched Posts:
Title: sunt aut facere repellat provident occaecati excepturi o
ptio 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
```

 Marwadi University Marwadi Chandarana Group 	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: 92400133055

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:
```

```
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}")
```

 Marwadi University Marwadi Chandarana Group 	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: 92400133055

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
JSONPlaceholderAPIClient:
    BASE_URL = "https://jsonplaceholder.typicode.com"
    def
__init__(self):
    pass
    def
get_posts(self):
    """Fetches a list of posts from the JSONPlaceholder API."""
    url = f"{self.BASE_URL}/posts"
    response = requests.get(url)
    if
response.status_code == 200:
        return response.json()
    else:
response.raise_for_status()
    def create_post(self, title, body,
user_id):
    """Creates a new post in the JSONPlaceholder API."""
    url = f"{self.BASE_URL}/posts"
    post_data = {
        "title": title,
        "body": body,
        "userId": user_id
    }
    response = requests.post(url, json=post_data)
    if
response.status_code == 201:
        return response.json()
    else:
response.raise_for_status()

# Example usage if
__name__ == "__main__":
```

 Marwadi University Marwadi Chandarana Group	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: 92400133055

```

client = JSONPlaceholderAPIClient()

# Fetching posts

try:
    posts = client.get_posts()
    print("Fetched Posts:")
    for post in posts[:5]: # Print first 5 posts
        print(post)
except Exception as e:
    print(f"Error fetching posts: {e}")

# Creating a new post
try:
    new_post = client.create_post("My New Post",
    "This is the content of my new post.",
    1)
    print("Created Post:")
    print(new_post)
except Exception as e:
    print(f"Error creating post: {e}")

```

Output:

```

Fetched Posts:
{'userId': 1, 'id': 1, 'title': 'sunt aut facere repellat provi
dent occaecati excepturi optio reprehenderit', 'body': 'quia et
suscipit\nsuscipit recusandae consequuntur expedita et cum\nre
prehenderit molestiae ut ut quas totam\nnostrum rerum est autem
sunt rem eveniet architecto'}
{'userId': 1, 'id': 2, 'title': 'qui est esse', 'body': 'est re
rum tempore vitae\nsequi sint nihil reprehenderit dolor beatae
ea dolores neque\nfugiat blanditiis voluptate porro vel nihil m
olestiae ut reiciendis\nqui aperiam non debitis possimus qui ne
que nisi nulla'}
{'userId': 1, 'id': 3, 'title': 'ea molestias quasi exercitation
nem repellat qui ipsa sit aut', 'body': 'et iusto sed quo iure\
nvoluptatem occaecati omnis eligendi aut ad\nvoluptatem dolorib
us vel accusantium quis pariatur\nmolestiae porro eius odio et
labore et velit aut'}
{'userId': 1, 'id': 4, 'title': 'eum et est occaecati', 'body':
'ullam et saepe reiciendis voluptatem adipisci\nsit amet autem
assumenda provident rerum culpa\nquis hic commodi nesciunt rem
tenetur doloremque ipsam iure\nquis sunt voluptatem rerum illo
velit'}
{'userId': 1, 'id': 5, 'title': 'nesciunt quas odio', 'body': '
repudiandae veniam quaerat sunt sed\nalias aut fugiat sit autem
sed est\nvoluptatem omnis possimus esse voluptatibus quis\nest
aut tenetur dolor neque'}
Created Post:
{'title': 'My New Post', 'body': 'This is the content of my new
post.', 'userId': 1, 'id': 101}

```