

"""

Python API & JSON Handling Script

Author: Your Name

Description:

This script fetches data from a public API,
parses JSON response, applies filtering logic,
and handles API errors properly.

"""

```
import requests # Library used to send HTTP requests
```

```
# API URL
```

```
API_URL = "https://jsonplaceholder.typicode.com/posts"
```

```
try:
```

```
    # Sending GET request to API
```

```
    response = requests.get(API_URL, timeout=5)
```

```
    # Raise error if status is not 200
```

```
    response.raise_for_status()
```

```
    print("API Request Successful!\n")
```

```
    # Convert response into JSON format (list of dictionaries)
```

```
    data = response.json()
```

```
# -----
# Filtering Logic Example
# -----


print("Posts by User ID = 1:\n")

filtered_posts = [post for post in data if post["userId"] == 1]

for post in filtered_posts[:5]: # Display first 5 posts

    print(f"Post ID: {post['id']}")

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

    print("-" * 40)

except requests.exceptions.HTTPError as errh:

    print("HTTP Error:", errh)

except requests.exceptions.ConnectionError:

    print("Error Connecting to API")

except requests.exceptions.Timeout:

    print("Request Timed Out")

except requests.exceptions.RequestException as e:

    print("Something went wrong:", e)

except Exception as e:
```

```
print("Unexpected Error:", e)
```

Github repository

<https://github.com/Karan6165/Alfido-Tech-Internship.git>

The screenshot shows the Microsoft Visual Studio Code interface with a Python script named `API_script.py` open in the editor. The code is a Python script that fetches data from a public API, parses JSON response, applies filtering logic, and handles API errors properly. It uses the `requests` library to send HTTP requests and the `json` module to convert the response into JSON format (list of dictionaries). The script then prints posts by User ID = 1 and filters posts where `post["userId"] == 1`. It handles various exceptions like `HTTPError`, `ConnectionError`, `Timeout`, `RequestException`, and `Exception`.

```
File Edit Selection View Go Run Terminal Help ← → Search API_script.py X D: > project > alrido tasks > task 2 > API_script.py > ... 1 """ 2 Python API & JSON Handling Script 3 Author: Your Name 4 Description: 5 This script fetches data from a public API, 6 parses JSON response, applies filtering logic, 7 and handles API errors properly. 8 """ 9 10 import requests # Library used to send HTTP requests 11 12 # API URL 13 API_URL = "https://jsonplaceholder.typicode.com/posts" 14 15 try: 16     # Sending GET request to API 17     response = requests.get(API_URL, timeout=5) 18 19     # Raise error if status is not 200 20     response.raise_for_status() 21 22     print("API Request Successful!\n") 23 24     # Convert response into JSON format (list of dictionaries) 25     data = response.json() 26 27     # Filtering Logic Example 28     print("Posts by User ID = 1:\n") 29 30     filtered_posts = [post for post in data if post["userId"] == 1] 31 32     for post in filtered_posts[:5]: # Display first 5 posts 33         print(f"Post ID: {post['id']}") 34         print(f"Title: {post['title']}") 35         print("-" * 40) 36 37 except requests.exceptions.HTTPError as errh: 38     print("HTTP Error:", errh) 39 40 except requests.exceptions.ConnectionError: 41     print("Error Connecting to API") 42 43 except requests.exceptions.Timeout: 44     print("Request Timed Out") 45 46 except requests.exceptions.RequestException as e: 47     print("Something went wrong:", e) 48 49 except Exception as e: 50     print("Unexpected Error:", e)
```

The terminal tab shows the output of running the script:

```
PROBLEMS OUTPUT DEBUG CONSOLE PORTS TERMINAL
● PS C:\Users\karan\AppData\Local\Programs\Microsoft VS Code> & C:\Users\karan\AppData\Local\Programs\Python\Python313\python.exe "d:/project/alfido tasks/task 2/API_script.py"
API Request Successful!

Posts by User ID = 1:

Post ID: 1
Title: sunt aut facere repellat provident occaecati excepturi optio reprehenderit
-----
Post ID: 2
Title: qui est esse
-----
Post ID: 3
Title: ea molestias quasi exercitationem repellat qui ipsa sit aut
-----
Post ID: 4
Title: eum et est occaecati
-----
Post ID: 5
Title: nesciunt quas odio
-----
```