**ANSWER**

**Part 1: Understanding APIs and REST APIs**

**Task 1: Research & Short Answers**

1. **What is an API?**
   * Provide a definition and give two real-world examples.
2. **What is a REST API?**
   * Explain the key principles (HTTP methods, statelessness, resources).
3. **List the common HTTP methods used in REST APIs and their purposes.**
   * (GET, POST, PUT, DELETE, PATCH)
4. **What is JSON? Why is it commonly used in APIs.**

**1.Ans**:- An API (Application Programming Interface) is a set of rules and protocols that allow one software application to interact with another. APIs enable different systems to communicate without needing to understand each other’s internal logic.

Example:-1.Google Maps API – Used by apps like Uber to show maps and directions.

2.Twitter API – Used by apps to post tweets or fetch timelines.

**2.Ans:-** A REST (Representational State Transfer) API is a type of web API that uses HTTP requests to access and manipulate data. RESTful APIs treat data as *resources*, and each resource is accessed via a unique URL.

Key principles:

* HTTP methods (GET, POST, PUT, DELETE)
* Statelessness – Each request contains all information needed; server doesn’t store session info.
* Resources – Data is organized around resources, represented in formats like JSON

**3.Ans:-** GET – Retrieve data.

* POST – Create new data.
* PUT – Update existing data (replaces entire resource).
* DELETE – Remove data.
* PATCH – Partially update a resource.

**4.Ans**:- JSON (JavaScript Object Notation) is a lightweight, human-readable data format used to exchange data between a client and server.

Why it’s used:

* Easy to read and write.
* Supported by most programming languages.
* Works well with JavaScript (common on the web).

**Part 2: Making API Requests with Python**

**Task 3: GET Request (Fetching Data)**

Use the JSONPlaceholder API (a free fake API for testing):

* Endpoint: https://jsonplaceholder.typicode.com/posts

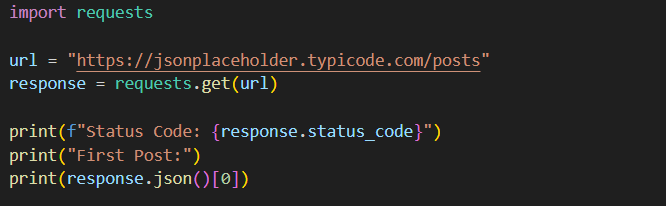
**Instructions:**

1. Write a Python script to fetch all posts from the API.
2. Print the response status code.
3. Print the first post in the response (JSON format).

**Expected Output:**

Status Code: 200

First Post: {'userId': 1, 'id': 1, 'title': '...', 'body': '...'}

****

**Task 4: POST Request (Sending Data)**

**Instructions:**

1. **Use the same API to create a new post.**
2. **Send a JSON payload with:**

{

"title": "New Post",

"body": "This is a test post.",

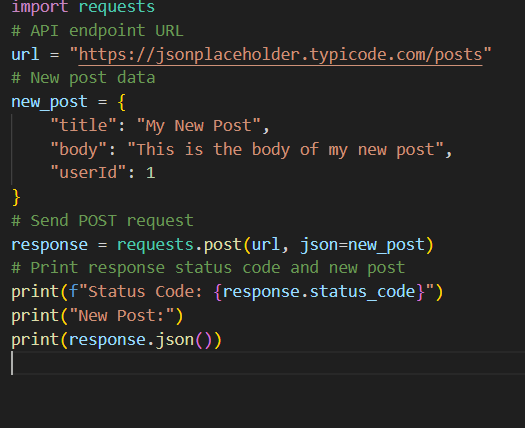
"userId": 1

}

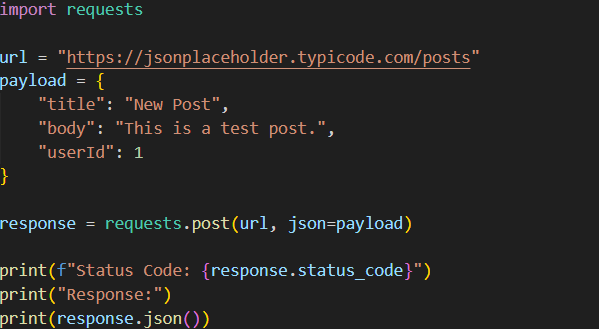
1. **Print the response (should include the new post with an ID).**

Expected Output:

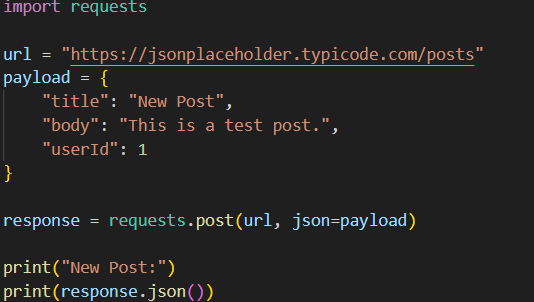
New Post: {'title': 'New Post', 'body': 'This is a test post.', 'userId': 1, 'id': 101}

**1.Ans:-**

**2.Ans:-**

****

**3.Ans:-**

****

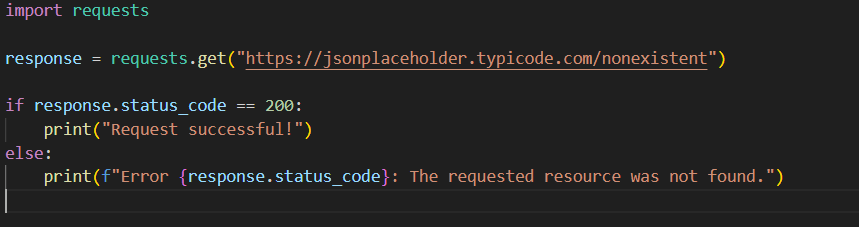
**Task 5: Error Handling & Authentication**

**Instructions:**

1. Try accessing a non-existent endpoint (e.g., https://jsonplaceholder.typicode.com/nonexistent).
2. Handle the error (check status code, print an error message if request fails).

**Example Code:**

response = requests.get("https://jsonplaceholder.typicode.com/nonexistent")



**Task 6: Fetch Weather Data (OpenWeatherMap API)**

1. Sign up for a free API key at [OpenWeatherMap](https://openweathermap.org/api).
2. Fetch the current weather for a city of your choice.
3. Print temperature, weather description, and humidity.

**Endpoint:**

[https://api.openweathermap.org/data/2.5/weather?q={city}&appid={API\_KEY}&units=metric](https://api.openweathermap.org/data/2.5/weather?q=%7bcity%7d&appid=%7bAPI_KEY%7d&units=metric)

