



# Communicating with APIs

By GS APWE

5 June 2025



# What is an API?

- Application Programming Interface
- Allows programs to talk to each other
- The messenger between client and server

# Why are APIs useful?

- Access real-time data (weather, news, music, etc.)
- Automate tasks
- Let your application integrate with other applications (Instagram, Spotify, etc.)

<https://www.botdistrikt.com/busuncle>



# Types of APIs

1. **REST API:** most common
  - Every piece of data lives at a specific URL (endpoint)
  - Response is pre-defined
  - <https://accuweather.com/weather?location=Singapore>
2. **GraphQL:** newer, more efficient, but also more complex
  - One URL for all data, but need to modify the request body
  - Response is customized by you
  - <https://accuweather.com/graphql>

We focus on REST API in this workshop.

# Key Concepts in REST API

- Endpoint: URL to access the resource
- Query methods:
  1. GET (retrieve data)
  2. POST (send new data)
  3. PUT (update data)
  4. DELETE (remove data)
- Headers: contain metadata (such as API key)
- Parameters: extra information to the query
- Response: usually in JSON format

# Fun Example

[http://www.official-joke-api.appspot.com/random\\_joke](http://www.official-joke-api.appspot.com/random_joke)



The screenshot shows a code editor with a dark theme. The top bar includes a 'Personal' dropdown and a 'Version control' dropdown. The left sidebar shows a 'Project' view with a folder named 'Personal' containing a file 'Main.py'. The main editor area shows the following Python code:

```
1 import requests
2 response = requests.get("https://official-joke-api.appspot.com/random_joke")
3 print(response.json())
```

Below the editor is a 'Run' section with a 'Main' tab. It shows the command used to run the script:

```
/Users/hanguyen/opt/anaconda3/envs/systematic-trading/bin/python /Users/hanguyen/Library/CloudStorage/OneDrive-GeorgiaInstituteofTechnology/Personal/Main.py
```

The output of the script is displayed below the command:

```
{'type': 'general', 'setup': 'What did the scarf say to the hat?', 'punchline': 'You go on ahead, I am going to hang around a bit longer.', 'id': 183}
```

At the bottom, it states 'Process finished with exit code 0'.

# Rate Limits & Other Errors

- APIs don't allow unlimited requests
- Rate limit: how many requests you can make in a certain time
  - For example: 60 requests per minute
- If you exceed the limit (by sending too many requests), the API might block you
- Other common errors:
  - 400 – Bad Request: typo or wrong request format
  - 404 – Not Found: the endpoint doesn't exist
  - 500 – Internal Server Error: server is having problems

# Response Handling Example

---

```
Main.py x
1 import requests
2 response = requests.get("https://official-joke-api.appspot.com/random_joke")
3 if response.status_code == 200:
4     print(response.json())
5 else:
6     print("Something went wrong:", response.status_code)
```





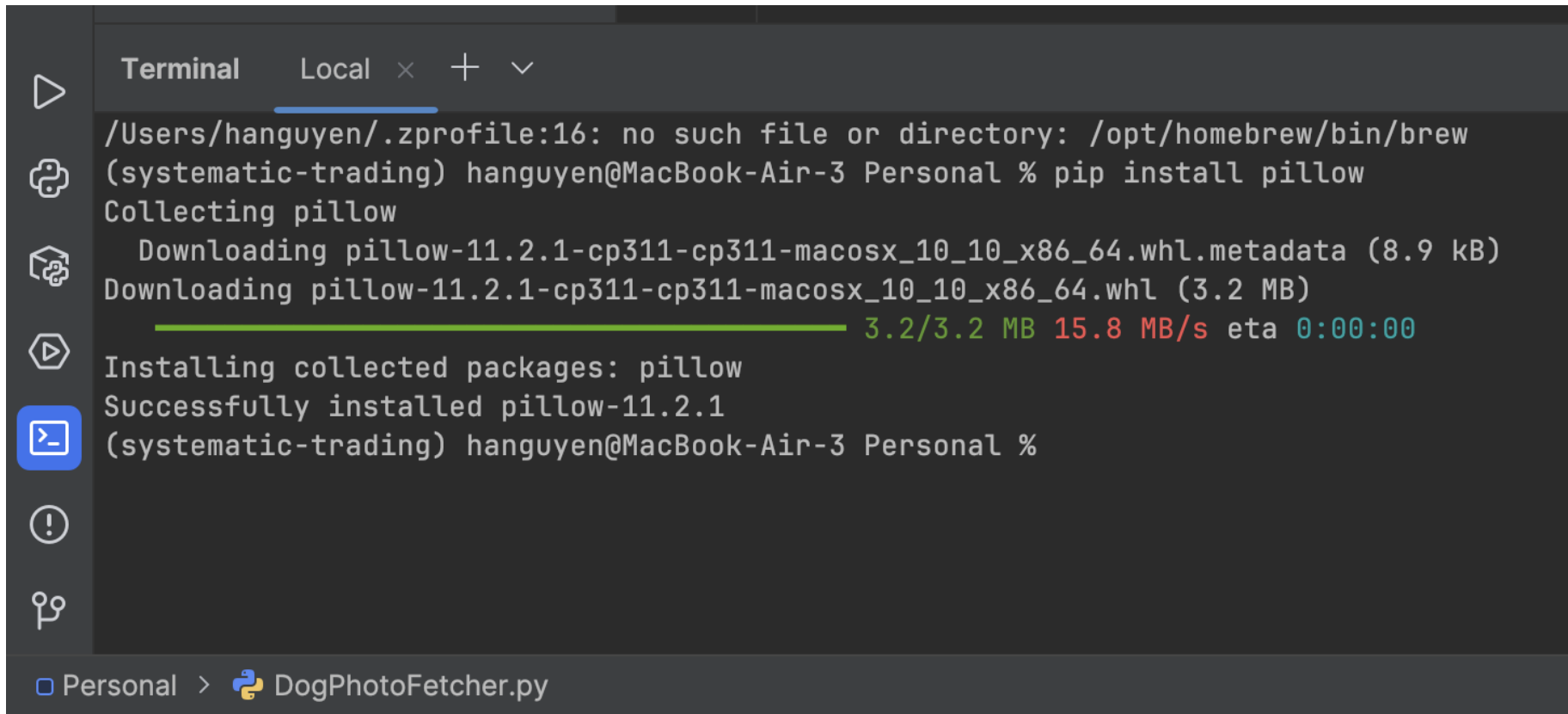
# Challenge Time

---

Can you get a random dog photo using this URL and display it?

<https://dog.ceo/api/breeds/image/random>

# Pre-requisite



A screenshot of a macOS Terminal window. The window has a dark gray background and a light gray title bar. The title bar contains the word "Terminal" and a tab labeled "Local" with a close button (x) and a dropdown arrow (v). The terminal content shows the execution of the command `pip install pillow`. The output indicates that the package is being collected, downloaded (3.2 MB at 15.8 MB/s), and successfully installed. The prompt shows the user is in a directory named "Personal" and the file "DogPhotoFetcher.py" is open in the background.

```
Terminal Local x + v
/Users/hanguyen/.zprofile:16: no such file or directory: /opt/homebrew/bin/brew
(systematic-trading) hanguyen@MacBook-Air-3 Personal % pip install pillow
Collecting pillow
  Downloading pillow-11.2.1-cp311-cp311-macosx_10_10_x86_64.whl.metadata (8.9 kB)
  Downloading pillow-11.2.1-cp311-cp311-macosx_10_10_x86_64.whl (3.2 MB)
    3.2/3.2 MB 15.8 MB/s eta 0:00:00
Installing collected packages: pillow
Successfully installed pillow-11.2.1
(systematic-trading) hanguyen@MacBook-Air-3 Personal %
```

Personal > DogPhotoFetcher.py

# Solution

---

```
Main.py  DogPhotoFetcher.py x
1  import requests
2  from PIL import Image
3  from io import BytesIO
4
5  # Send a GET request to the API
6  url = "https://dog.ceo/api/breeds/image/random"
7  response = requests.get(url)
8
9  # Check if the request was successful
10 if response.status_code == 200:
11     data = response.json()
12     image_url = data['message']
13     print("Random Dog Image URL:", image_url)
14
15     # Download and display the image
16     image_response = requests.get(image_url)
17     img = Image.open(BytesIO(image_response.content))
18     img.show()
19 else:
20     print("Failed to fetch dog image. Status code:", response.status_code)
21
```

# Real World Applications



## Social Media

Content creators can use TikTok API to schedule posts



## Map & Transport Apps

Google Maps use APIs to plan your route, calculate ETA

TADA and Grab use APIs to check for routes and available drivers



## Music

Connect with Spotify API to build a music search tool

# Using Spotify API

- Go to <https://developer.spotify.com/dashboard>
- Log in with your account
- Click “Create App”
- Fill in the details
- Copy your Client ID and Client Secret
- Use the sample code to run your song search application



Try out Telegram Bot now!