

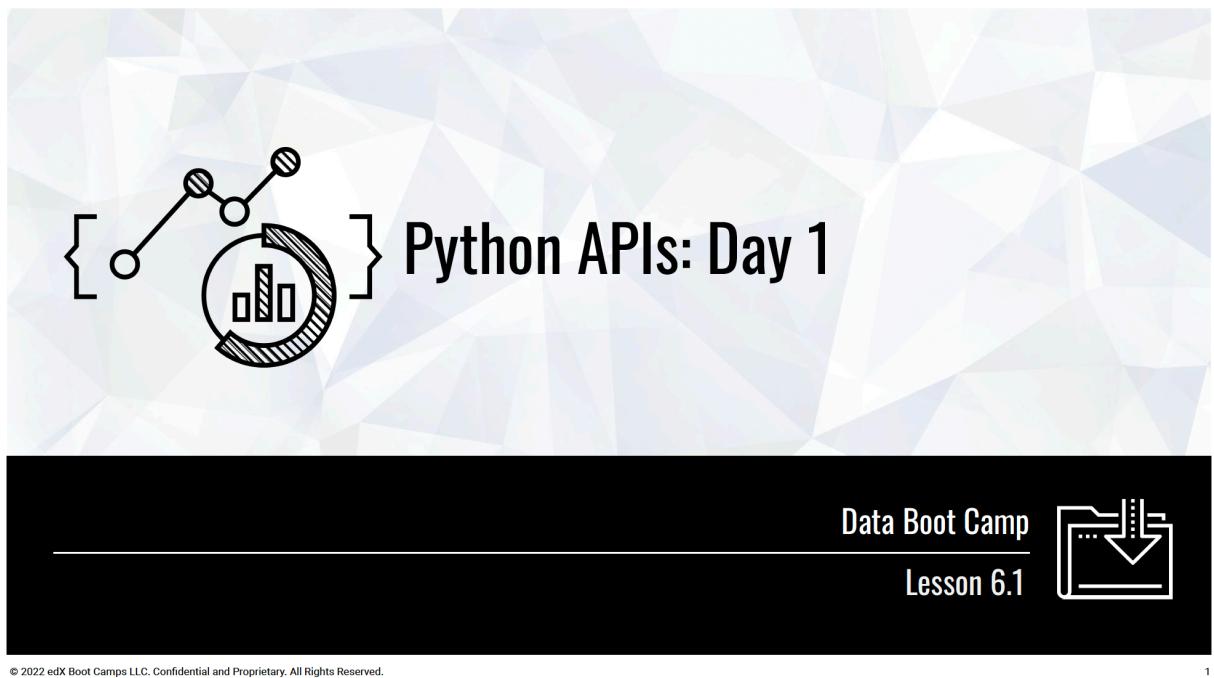
# MOD 6 DAY 1 API'S

Tuesday, October 17, 2023      6:10 PM

### WHAT IS AN API

### WHAT IS A JSON

### what is a client vs server



## Class Objectives

By the end of today's class, you will be able to:



Make get requests with Python's Requests library

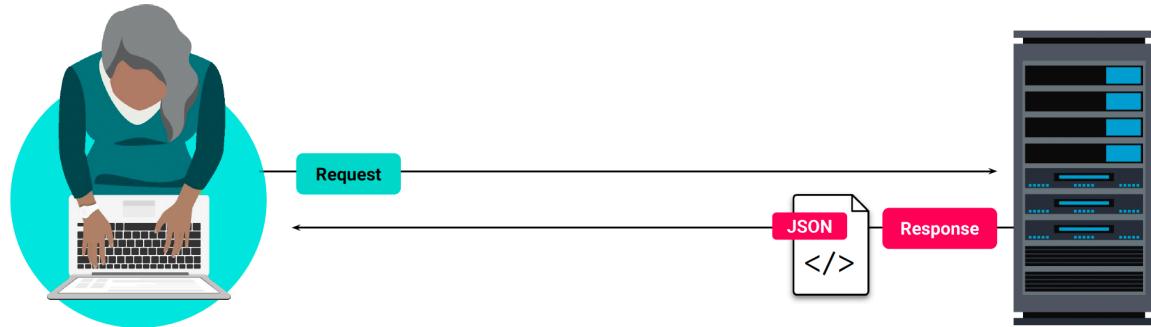


- Manipulate JSON responses to retrieve necessary values
- Store JSON responses in Python lists and dictionaries
- Use the OMDb API documentation to create requests for movie data

2

## What Is a Client versus a Server?

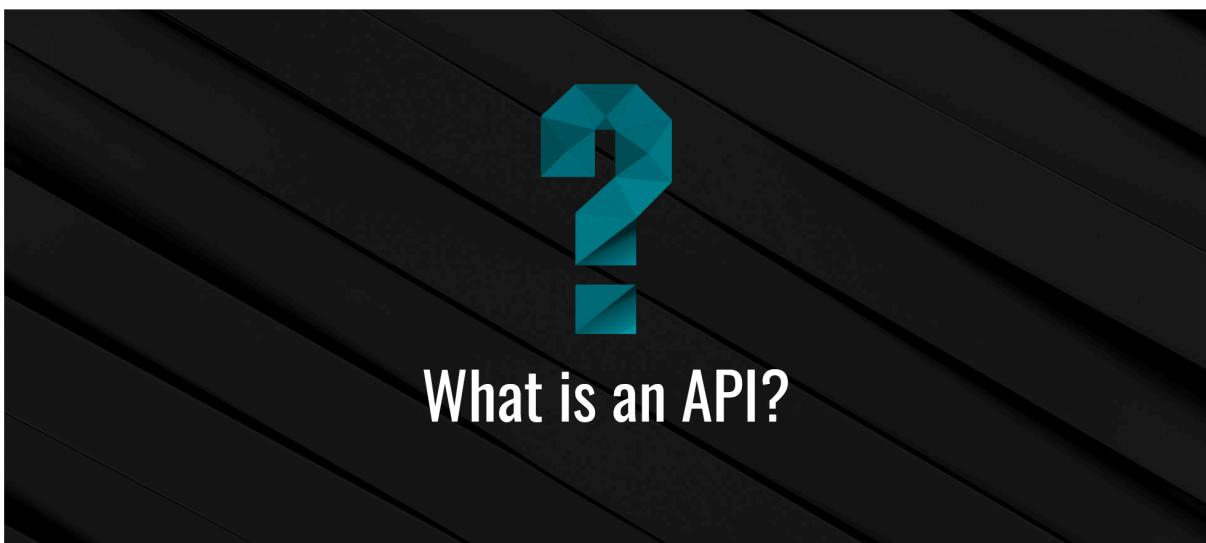
**Analogy:** A patient asks a health question, and a doctor supplies the answer.



A **client** is an application or device that asks for information.

A **server** is an application or device that supplies information to the client.

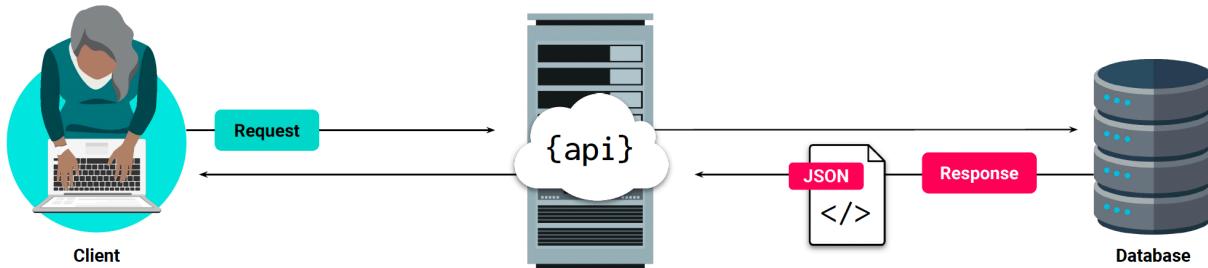
3





## Application Programming Interface (API)

-  A request is a communication to the API to retrieve data.
-  API calls are similar to visiting a website in a browser.
-  They point to a URL and collect some data from the page.



## JavaScript Object Notation (JSON)

-  A webpage may return a JSON in response to an API call.
-  The URLs used to communicate with APIs are called endpoints.
-  The text in the web browser is identical to what a client script would receive.

```
[
  {
    "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"
  }
]
```



## Activity 1

00:09:00

Request intro

### what is a json

### what is an api

Import dependencies

How to print object responses

How to convert api into json





## There Are Two Components to Our API Request

01

`requests.get(url)` Sends a get request to the URL, passed as a parameter.

```
# Dependencies
import requests
import json
```

```
# URL for GET requests to retrieve vehicle data
url = "https://api.spacexdata.com/v2/launchpads"
```

```
# Print the response object to the console
print(requests.get(url))
```

8

## There Are Two Components to Our API Request

02

A call to convert the response object into a JSON format.

`json.dumps()` is a method used to “pretty print” the response.

```
# Pretty Print the output of the JSON
response = requests.get(url).json()
print(json.dumps(response, indent=4, sort_keys=True))

[{
    "details": "SpaceX primary Falcon 9 launch pad, where all east coast Falcon 9s launch ed prior to the AMOS-6 anomaly. Initially used to launch Titan rockets for Lockheed Martin. H eavily damaged by the AMOS-6 anomaly with repairs expected to be complete by late summer 201 7.",
    "full_name": "Cape Canaveral Air Force Station Space Launch Complex 40",
    "id": "ccafs_slc_40",
    "location": {
        "latitude": 28.5618571,
        "longitude": -80.577366,
        "name": "Cape Canaveral",
        "region": "Florida"
    },
    "status": "under construction",
    "vehicles_launched": "falcon 9"
},
{
    "details": "SpaceX new launch site currently under construction to help keep up with
the Falcon 9 and Heavy manifest. Expected to be completed in late 2018. Initially will be li
```

9



0:28:00

## Space x request

Import dependencies

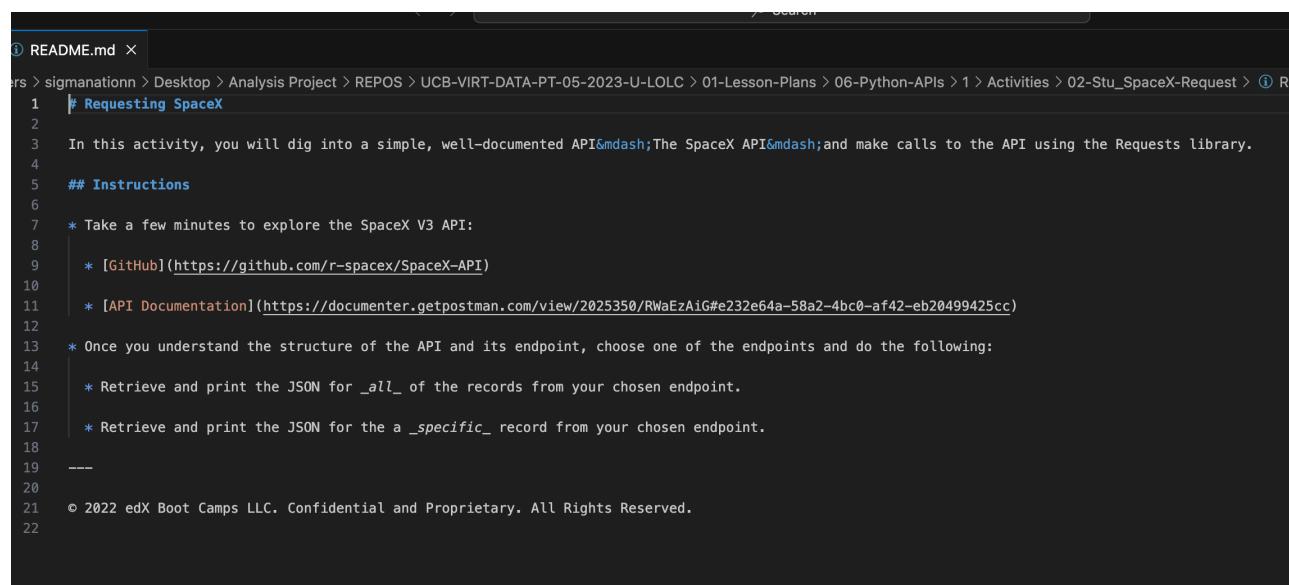
### pretty print

We pretty print

Pretty print json

We get a url request

### not all websites or api will be free



The screenshot shows a terminal window with a dark theme. The title bar says "README.md". The content of the file is as follows:

```
1 # Requesting SpaceX
2
3 In this activity, you will dig into a simple, well-documented API—the SpaceX API—and make calls to the API using the Requests library.
4
5 ## Instructions
6
7 * Take a few minutes to explore the SpaceX V3 API:
8
9     * [GitHub](https://github.com/r-spacex/SpaceX-API)
10
11    * [API Documentation](https://documenter.getpostman.com/view/2025350/RWaEzAiG#e232e64a-58a2-4bc0-af42-eb20499425cc)
12
13 * Once you understand the structure of the API and its endpoint, choose one of the endpoints and do the following:
14
15     * Retrieve and print the JSON for all of the records from your chosen endpoint.
16
17     * Retrieve and print the JSON for a specific record from your chosen endpoint.
18
19 ---
20
21 © 2022 edX Boot Camps LLC. Confidential and Proprietary. All Rights Reserved.
22
```

## Activity 3

30:00 (mis checked update later)



## Manipulation responses

### pretty print

### what does f' mean in python

## Activity 4

Far far away api

00:57:00

### Dependencies

Retrieve data from api

We create a character id

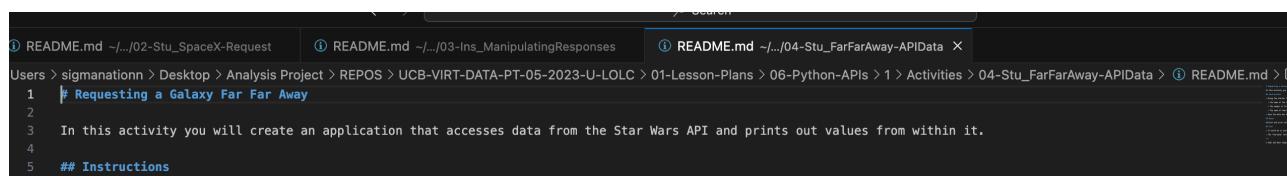
### read the json

We retrieve name from api json

we print a url as a response

### what is pretty printing

Print data within a json



The screenshot shows a terminal window with three tabs open, each containing a snippet of Python code:

- Tab 1: README.md (~.../02-Stu\_SpaceX-Request) - Contains a single line: `1 # Requesting a Galaxy Far Far Away`.
- Tab 2: README.md (~.../03-Ins\_ManipulatingResponses) - Contains a single line: `1 In this activity you will create an application that accesses data from the Star Wars API and prints out values from within it.`.
- Tab 3: README.md (~.../04-Stu\_FarFarAway-APIData) - Contains a single line: `1 ## Instructions`.



```
6 * Using the starter file provided, collect the following pieces of information from the Star Wars API.
7
8     * The name of the character
9
10    * The number of films they were in
11
12        * The name of their first starship
13
14    * Once the data has been collected, print it out to the console.
15
16
17 ## Bonus
18
19 Collect and print out all of the films a character appeared in.
20
21 ## Hint
22
23 * It would be in your best interest to print out the JSON from the initial request before anything else. This will let you know what keys you should reference.
24
25 * The "starship" values are links to another API call. This means that you will need to create a request based on the values of a previous request.
26
27 ---
28
29 © 2022 edX Boot Camps LLC. Confidential and Proprietary. All Rights Reserved.
30
```

## Activity 5

### Number facts

Import dependencue

Get url

### what is concatenate

### "?" means to quory json url

Numbersapi.com

.lower is to lower case the sting





an interactive application that uses the Numbers API.  
The application will take in a number and then return a random fact about that number.  
(Instructions sent via Slack.)

Suggested Time:

20 minutes

## Activity: Number Facts

Instructions	Using the Numbers API ( <a href="http://numbersapi.com">http://numbersapi.com</a> ), create an application that takes in a user's inputs and returns a number fact based on the inputs.
Hints	The URL to make your request to must have <code>?json</code> at its end so that the data format returned is JSON. The default response is pure text.  Make sure to read through the documentation when creating your application. Some types require more or less data than others.

## Activity 6

Omdb request

Omdbapi.com

Hugo\_92@hotmail.com  
Is hugos email

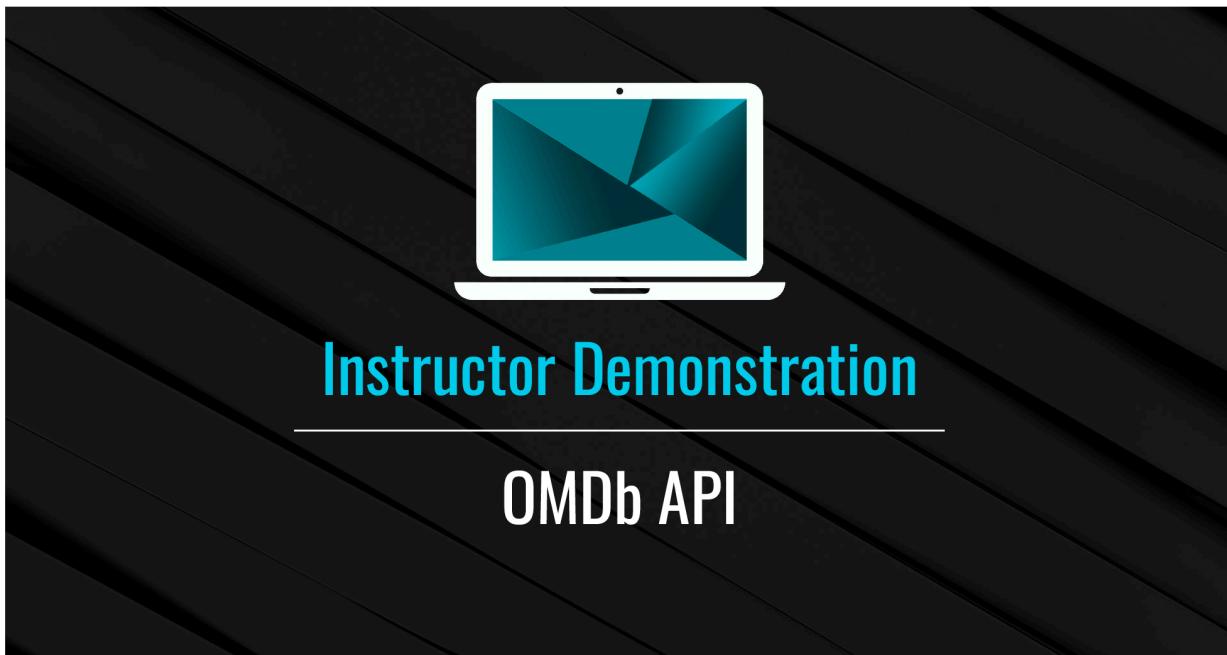
How to open an api key



1:57:00

Pprint (means pretty print )

Apis has movie keys 2:03:00 - 2:02:00



## JSON responses so far

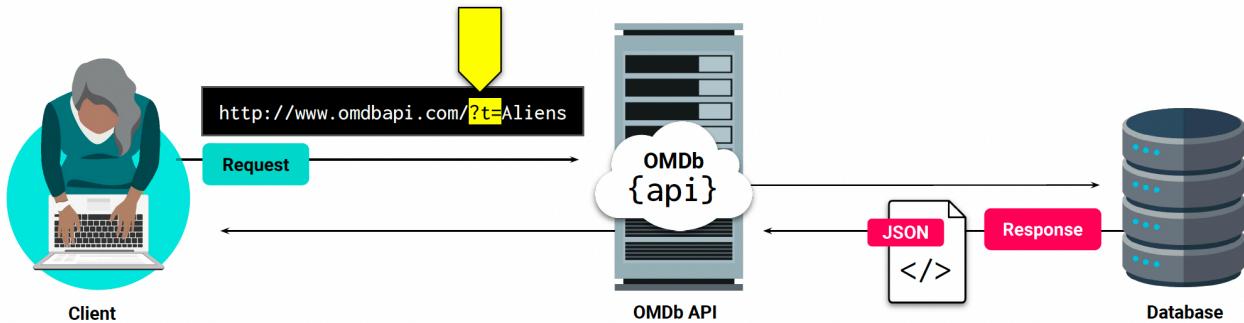
Our previous API responses have been pretty simple but the OMDb API is slightly more complex





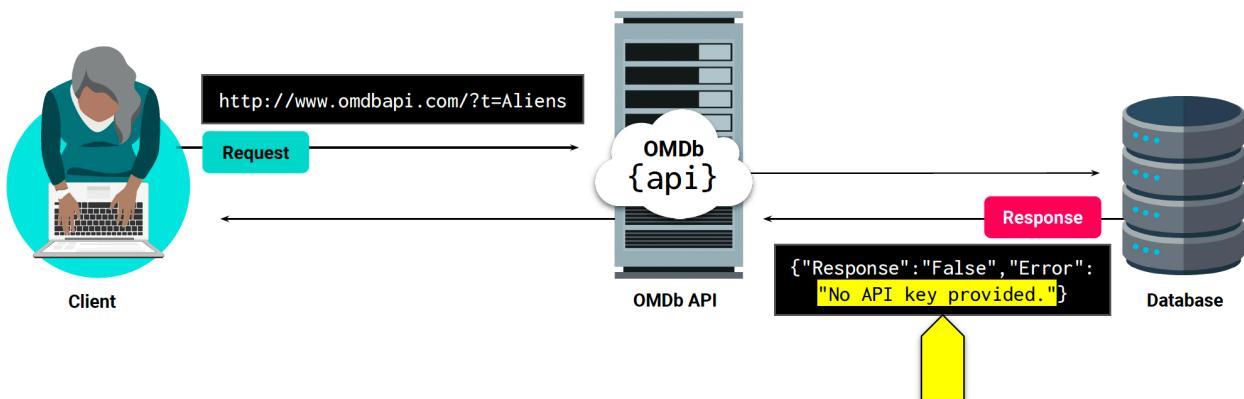
## URL Parameters: ?t=

This is asking the API to return all information on movies with the title “Aliens”.



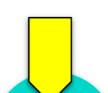
## URL Parameters: api\_key

Without an API key, no data would be returned.

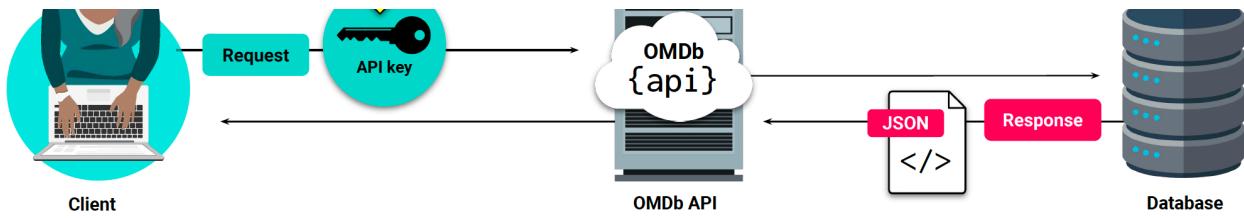


## URL Parameters: api\_key

API keys restrict API access to specific users.







## URL Parameters: `api_key`

---

```
# Note that the ?t= is a query param for title of the movie we want to search for
url = "http://www.omdbapi.com/?t="
api_key = "&apikey=trilogy"
```

```
# Performing a GET request similar to the one we executed earlier
response = requests.get(url + "Aliens" + api_key)
print(response.url)
```

<http://www.omdbapi.com/?t=Aliens&apikey=trilogy>

( I DINT NOTICE ACTIVITY 7)  
 (MUST BE BETWEEN 6 AND 8 )

Activity 8

Movie questions

2:19:-00

We impirt dependecie and url

He explains how to call api in a url

We pull specific data from the data in the url and print specific data





## Activity: Movie Questions

In this activity, you will test your skills with the OMDb API by collecting data to answer a series of questions.

(Instructions sent via Slack.)

Suggested Time:

20 minutes

## Activity: Movie Questions

Use the OMDb API to retrieve and print answers to the following questions:

Who was the director of the movie **Aliens**?

What was the movie **Gladiator** rated?

What year was **50 First Dates** released?

Who wrote **Moana**?

What was the plot of the movie **Sing**?

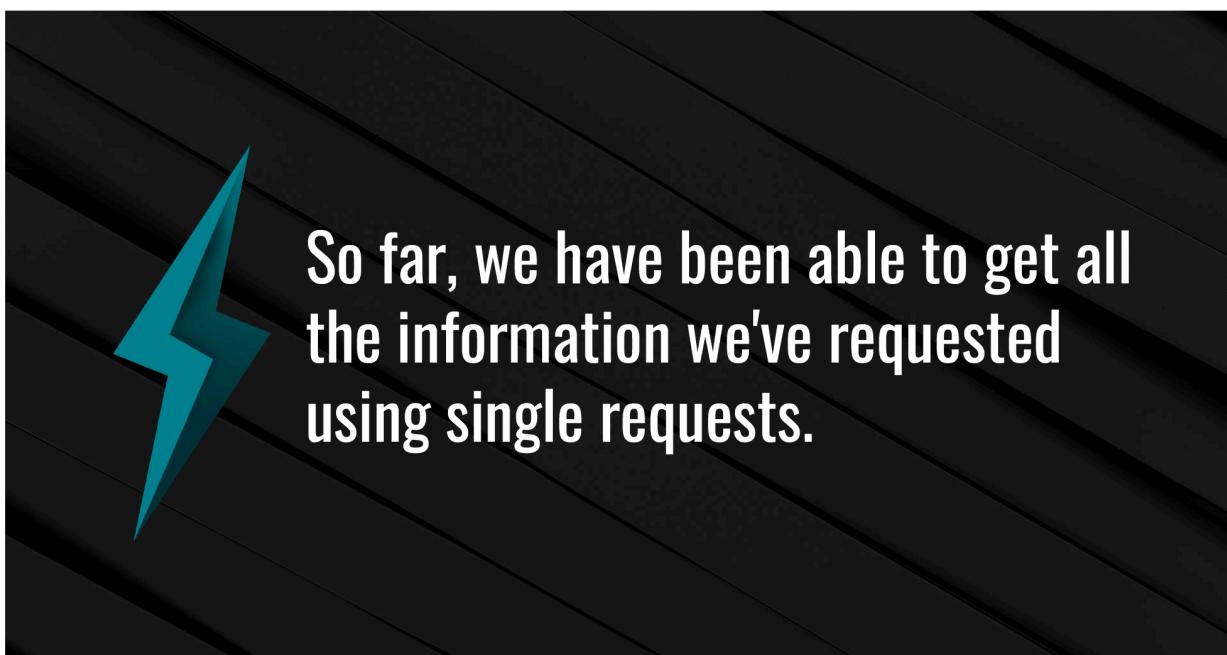


Aciticy 9

2:22:00

We collect 100 posts

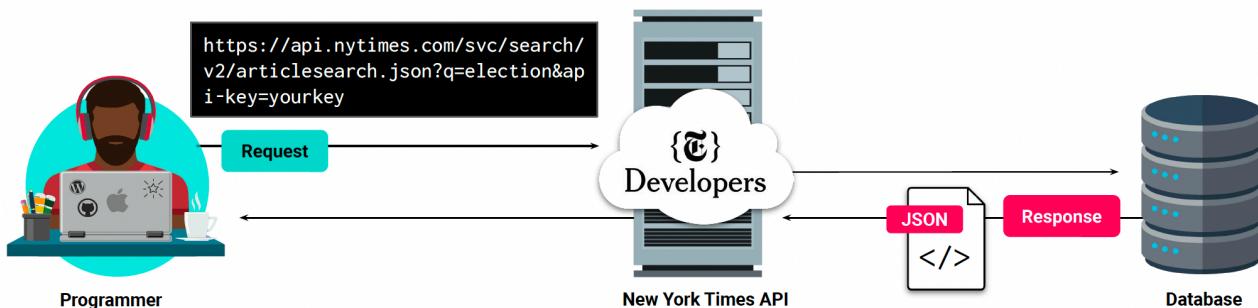
Create and eompy list to store data





## Our Requests So Far

- API keys restrict API access to specific users.
- Without an API key, no data would be returned.  
A programmer would have to make 5 requests to retrieve 50 articles.



## Requests on a Loop!

```
# Make a request for each of the indices
for x in range (len(indices)):
    print(f"Making request number: {x} for ID: {indices[x]}")

# Get one of the posts
post_response = requests.get(url + str(indices[x]))

# Save post's JSON
response_json.append(post_response.json())
```

```
Making a request number: 0 for ID: 30
Making a request number: 1 for ID: 54
Making a request number: 2 for ID: 44
Making a request number: 3 for ID: 76
Making a request number: 4 for ID: 46
Making a request number: 5 for ID: 75
Making a request number: 6 for ID: 69
Making a request number: 7 for ID: 40
Making a request number: 8 for ID: 48
Making a request number: 9 for ID: 5
```

## Activity: Iterative Requests

In this activity, you will test your knowledge of iterative requests by looping through a list of movies and collecting data on each movie from the OMDB API.



(Instructions sent via Slack.)

Suggested Time:

10 minutes

## Activity: Iterative Requests

### Instructions

Consider the following list of movie titles: `movies = ["Aliens", "Sing", "Moana"]`

Make a request to the OMDB API for each movie in the list. Then, do the following:

- Print the director of each movie.
- Save the responses in another list.

The director of Aliens was James Cameron.

The rating of Gladiator was R.

The movie 50 First Dates was released in 2004.

Moana was written by Jared Bush (screenplay by), Ron Clements (story by), John Musker (story by), Chris Williams (story by), Don Hall (story by), Pamela Ribon (story by), Aaron Kandell (story by), Jordan Kandell (story by).

The plot of Sing was: In a city of humanoid animals, a hustling theater impresario's attempt to save his theater with a singing competition becomes grander than he anticipates even as its finalists' find that their lives will never be the same..

## Activity 10

2:37:00

Movie loop

Import depencede

We concatonate a url and an apu

We creal a list of movies

We create a loop within the list within the url



~~WE CREATE A LOOP WITHIN THE LIST WITHIN THE URL~~

## Activity 11

2:40:00

NYT API

How to create an api and upload it to git with a gitignore

How to query an url for all urls containing granola

Build query url

## Activity 12

Retrieve articles

2:45:00



We search artice between a cetrtaind data and lock for a specific sting

We vew snipete from article

BONUS

We loop though 0 to 3

```
1  # Retrieving Articles
2
3  In this activity, you will create an application that grabs articles from the NYT API, stores them within a list, and prints snippets of the articles to the screen.
4
5  ## Instructions
6
7  * Save the NYT API endpoint to a variable. Make sure that you include the right query parameter for retrieving JSON data!
8
9  * Register for and save your API Key to a variable.
10
11 * Decide on a search term, and save it to a variable.
12
13 * Limit your search to articles published within a range of datess; for example, only articles published in 2014. Hint_: Read the documentation on `end_date`.
14
15 * Build your query URL, and save it to a variable.
16
17 * Retrieve a response from the NYT API with a get request.
18
19 * Review the documentation. How do you get a hold of the articles in the response?
20
21 * Traverse through the returned JSON to retrieve the list of articles and store it in a variable.
22
23 * Print a `snippet` from each article, and separate each snippet using dashes (`-`).
24
25 ## Bonus
26
27 Figure out how we could get 30 results. **Hint**: Look up the `page` query parameter. If you get a message saying you've exceeded your rate limit, don't frets; you've solved the problem.
28
29 ## Hint
30
31 * **Warning:** Be sure not to print out any of the query URLs. The query URLs will include your API key, and if pushed to a public repository, it becomes a security risk for you as someone could steal and use your key.
32
```







## Activity 7

07-Stu\_Explore\_OMDb\_API

In this activity, you'll review the OMDB API documentation, and you'll practice using the API.

(in class demo)

## Activity 8

## 08-Stu\_MovieQuestions

```
# Dependencies  
# Who was the director of the movie Aliens?  
# What was the movie Gladiator rated?  
# What year was 50 First Dates released?  
# Who wrote Moana?  
# What was the plot of the movie Sing?
```

## Activity 9

## 09-Ins\_IterativeRequests

## # Dependencies



```
.. --> [REDACTED]
# Let's get the JSON for 100 posts sequentially.
# Create an empty list to store the responses
# Create random indices representing
# a user's choice of posts
# Make a request for each of the indices
    # Get one of the posts
    # Save post's JSON
# Now we have 10 post objects,
# which we got by making 100 requests to the API.
# preview the json
```

## Activity 10

### 10-Stu\_MovieLoop

```
# Dependencies
# Make a request to the OMDb API for each movie in the list.
    # Print the director of each movie
    # Save the responses in another list
# Print the responses
```

## Activity 11

### 11-Ins\_NYTAPI

```
# Dependencies
# Search for articles that mention granola
# Build query URL
# Request articles
# The "response" property in articles contains the actual articles
# list comprehension.
# Print the web_url of each stored article
```



## Activity 12

### 12-Stu\_RetrieveArticles

```
# Dependencies
# Store a search term
# Search for articles published between a begin and end date
# Build URL
# Retrieve articles
# BONUS: How would we get 30 results?
# HINT: Look up the page query param

# Empty list for articles
# loop through pages 0-2
# create query with page number
# Add a one second interval between queries to stay within API query limits
# loop through the response and append each article to the list
```

