DATA SCIENCE

DREAM JOB

Python Web APIs

Table of Contents

1. Introduction

a. What is an API?
b. Why use API?
c. Pros/Cons of API
d. Client and Server Architecture

2. Request and Response

a. REST Architecture
b. Response (JSON)
c. How to read JSON (Key/Value)

3. Creating API Key

a. What is an API Key?
b. How do I get one?

4. REST API Breakdown

a. Three Parts of API Requests
b. Reading API Documentation
c. REST API Demonstration

5. Notebook Demo

d. Learn the OpenWeatherMap API
e. Learn how to extract data from JSON format
f. Learn how to query multiple API calls

6. Questions

What are APIs?

Application

Programming

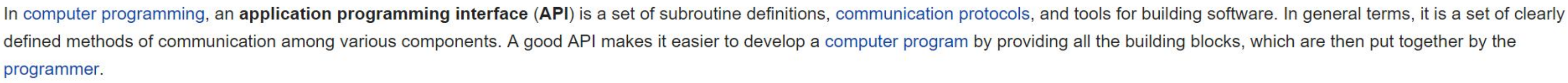
Interface

Or…

An **I**nterface
used by **P**rograms
to interact with an **A**pplication

What are APIs?

Wikipedia:



In computer programming, an **application programming interface** (**API**) is a set of subroutine definitions, communication protocols, and tools for building software. In general terms, it is a set of clearly defined methods of communication among various components. A good API makes it easier to develop a computer program by providing all the building blocks, which are then put together by the programmer.

"Application Programming Interfaces makes it easier for developers to use certain technologies in building applications" - **Wikipedia**

Source: https://en.wikipedia.org/wiki/Application_programming_interface

What are APIs?

In short...

API’s allow programs to **interact** and **communicate** with other programs.

API Application Programming Interface

It's a way for two pieces of software to talk to each other

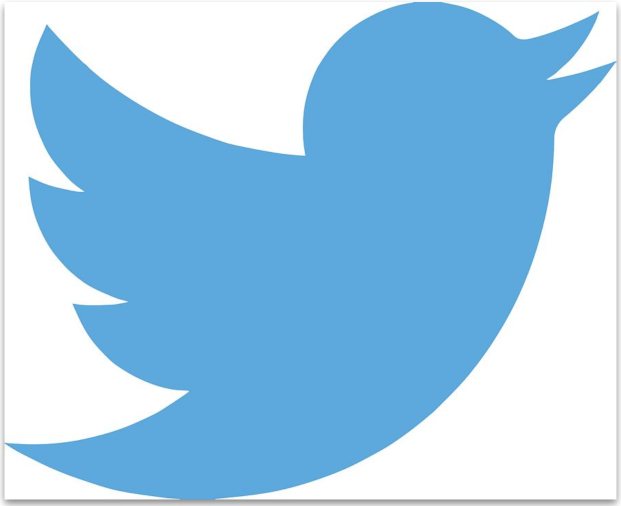
Web API’s

In the context of the web, these are known as **Web API's.** This allows you to send commands to programs running on the servers that you connect to from your browser.

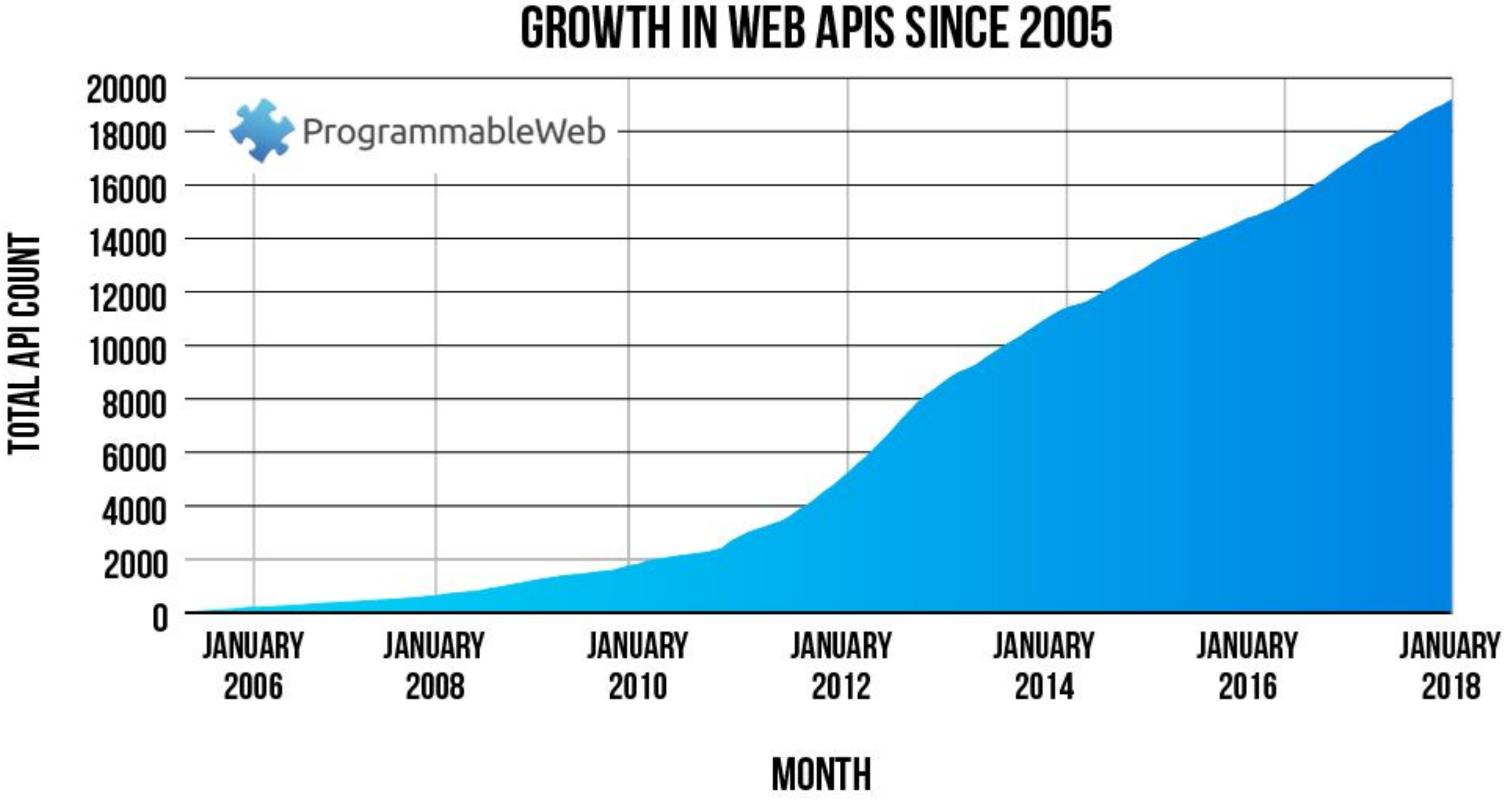
Web API’s are useful because:

- It **simplifies** data gathering for users and companies

Why are APIs so Important?



Growth of Web APIs

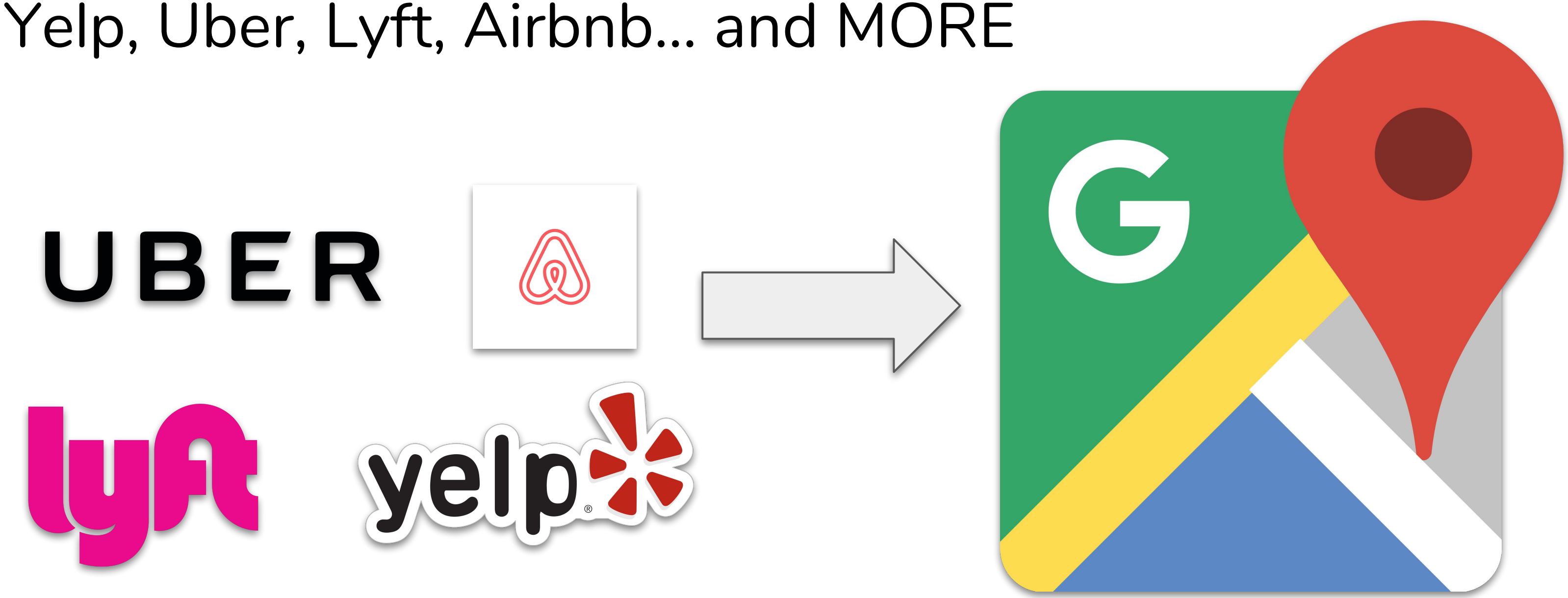


Source: https://www.programmableweb.com/sites/default/files/growth-in-web-apis-since-2005_0.png

Example of API Use Case

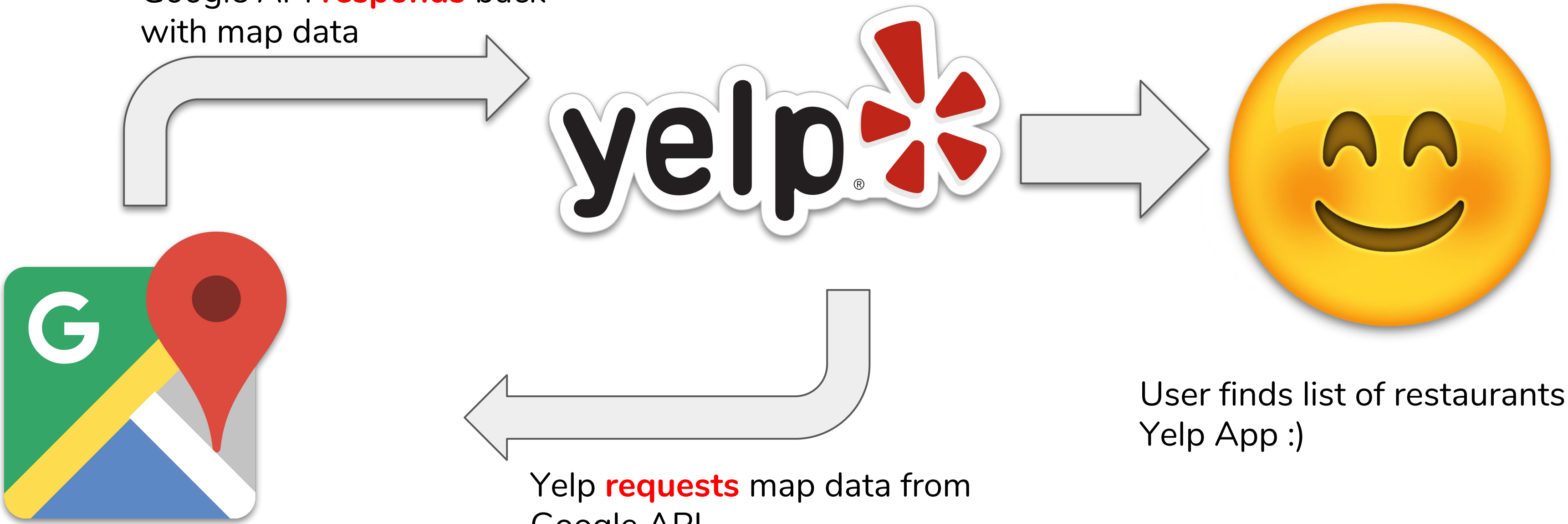
Google Maps API have been used by:

Yelp, Uber, Lyft, Airbnb... and MORE



Example: Yelp API

All of these APIs can help provide a service



Google API **responds** back with map data

Yelp **requests** map data from Google API...

User finds list of restaurants on Yelp App :)

API Pros/Cons

Pros: 💪

- ❑ **Time** - Saves companies lots of time because they don’t have to build anything in house
- ❑ **Cost** - It can save companies lots of money by using APIs
- ❑ **Focus** - Using an API allows companies to focus on their main product
- ❑ **Accessibility** - Nearly any programming language can access it with its libraries (Python, R, Java, Ruby, JavaScript…)

Cons: 😫

- ❑ **Security** - Since you’re using another company’s data, there is always a chance for security issues
- ❑ **Reliability** - Your product/service is accountable to the usage of their API. If it fails, then you’re at risk of failing as well.

What is REST?

REpresentational

These are the **resources** being provided and can be **represented** in different forms (JSON, XML, etc..)

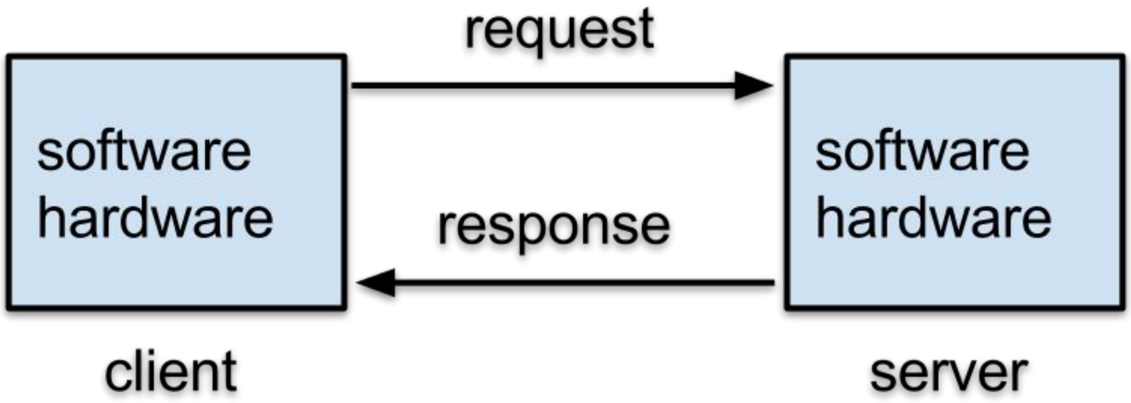
State

State transfer refers to a type of **transaction** on the resource (e.g. GET request)

Transfer

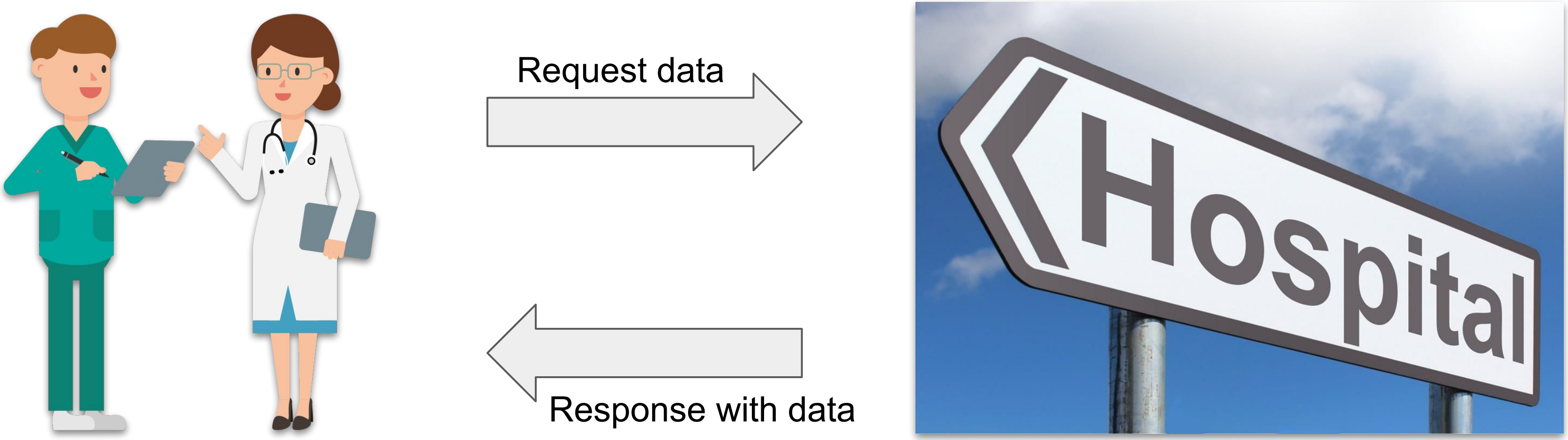
REST Client and Server Architecture

- ❏ **REQUEST** - Client is the application/device that ***requests*** for information.
- ❏ **RESPONSE** - Server is a computer that ***provides or responds back*** the information to the client.



REST Client and Server Example

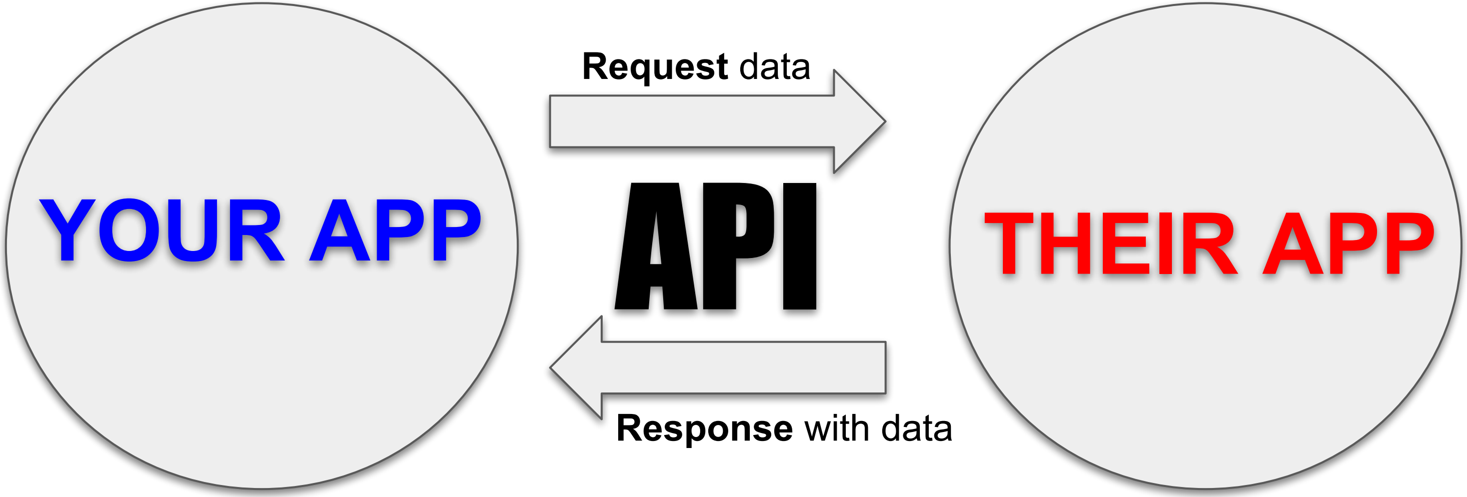
- The **doctors (client)** ask for patient medical records.
- The **hospital (server)** provides the necessary **information (data)**



Request data

Response with data

REST APIs Can Use Request and Response



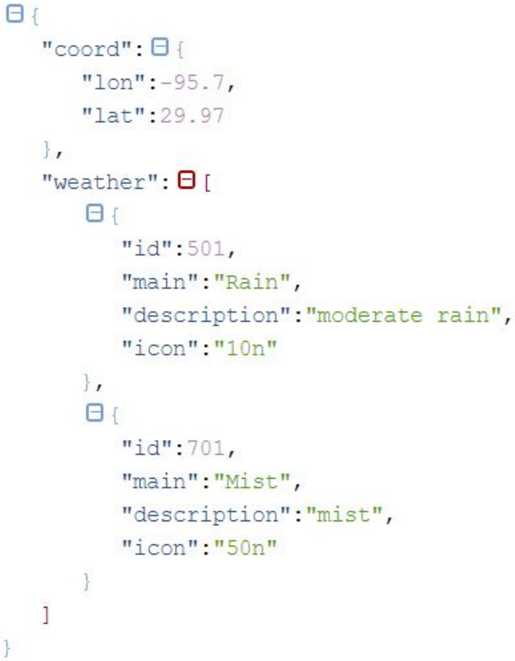
Python API - JSON Response

- When we want to interact with an API in Python (like accessing web services), it is very common to get the **responses** in a form called **JSON**.

JSON

JSON is very useful due to its hierarchical structure:

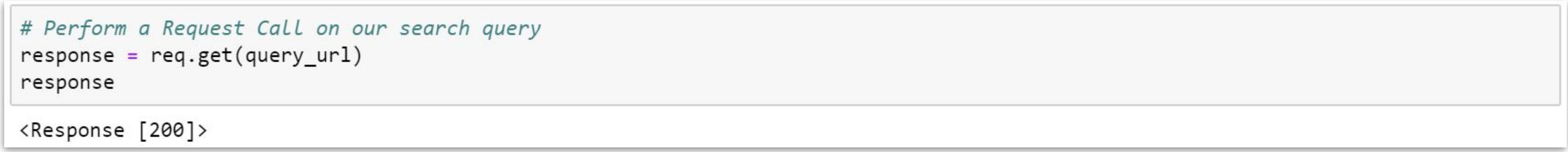
dictionary of dictionaries



```
⊟{
   "coord":⊟{
      "lon":-95.7,
      "lat":29.97
   },
   "weather":⊟[
      ⊟{
         "id":501,
         "main":"Rain",
         "description":"moderate rain",
         "icon":"10n"
      },
      ⊟{
         "id":701,
         "main":"Mist",
         "description":"mist",
         "icon":"50n"
      }
   ]
}
```

Request & Response

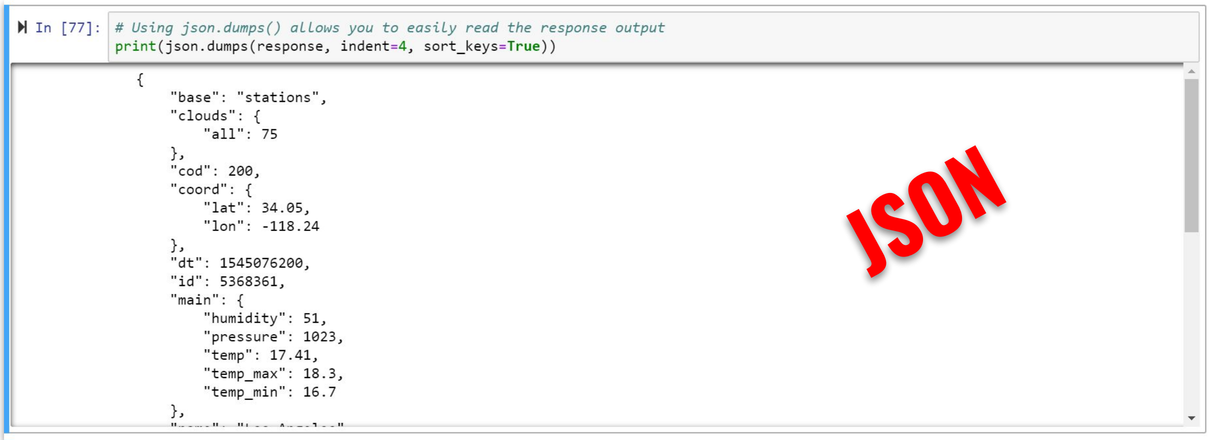
Request



```
# Perform a Request Call on our search query
response = req.get(query_url)
response
```

```
<Response [200]>
```

Response



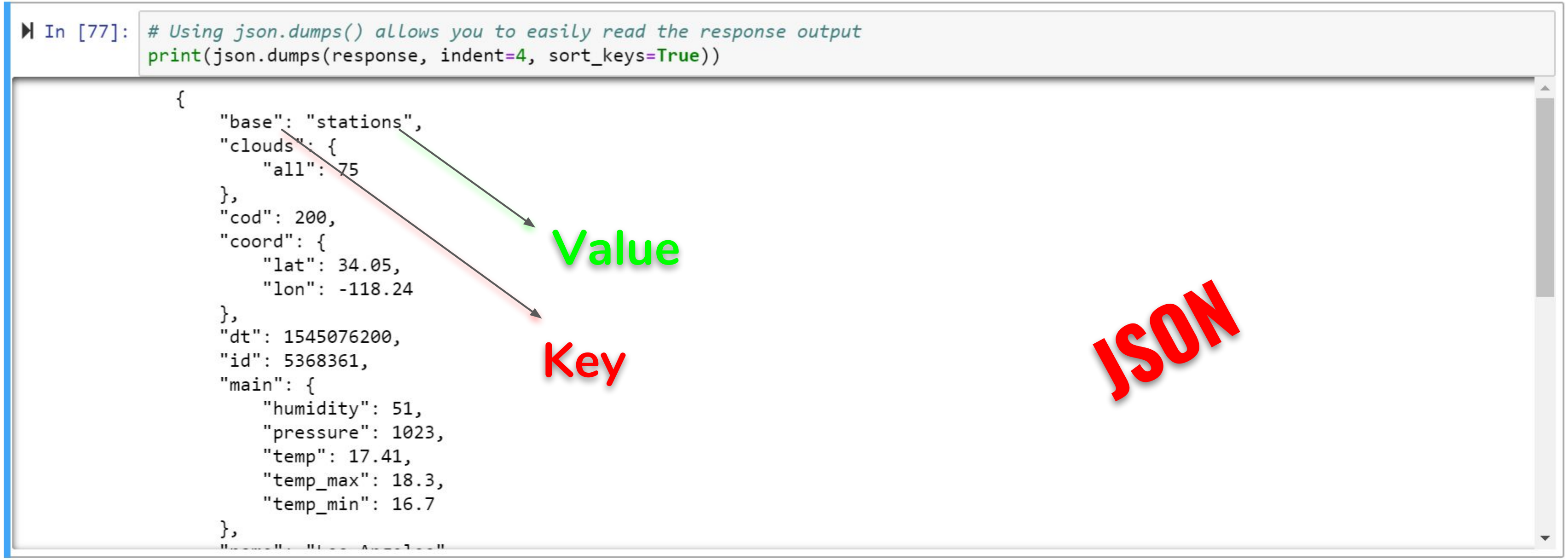
```
In [77]: # Using json.dumps() allows you to easily read the response output
         print(json.dumps(response, indent=4, sort_keys=True))
```

```
{
    "base": "stations",
    "clouds": {
        "all": 75
    },
    "cod": 200,
    "coord": {
        "lat": 34.05,
        "lon": -118.24
    },
    "dt": 1545076200,
    "id": 5368361,
    "main": {
        "humidity": 51,
        "pressure": 1023,
        "temp": 17.41,
        "temp_max": 18.3,
        "temp_min": 16.7
    },
    "name": "Los Angeles"
```

JSON

JSON Format

- ❑ **Similar to Key/Value Pair (Key : Value)**
- ❑ It’s just like a **dictionary** in Python



```
In [77]: # Using json.dumps() allows you to easily read the response output
         print(json.dumps(response, indent=4, sort_keys=True))
```

```
{
    "base": "stations",
    "clouds": {
        "all": 75
    },
    "cod": 200,
    "coord": {
        "lat": 34.05,
        "lon": -118.24
    },
    "dt": 1545076200,
    "id": 5368361,
    "main": {
        "humidity": 51,
        "pressure": 1023,
        "temp": 17.41,
        "temp_max": 18.3,
        "temp_min": 16.7
    },
    "name": "Los Angeles"
```

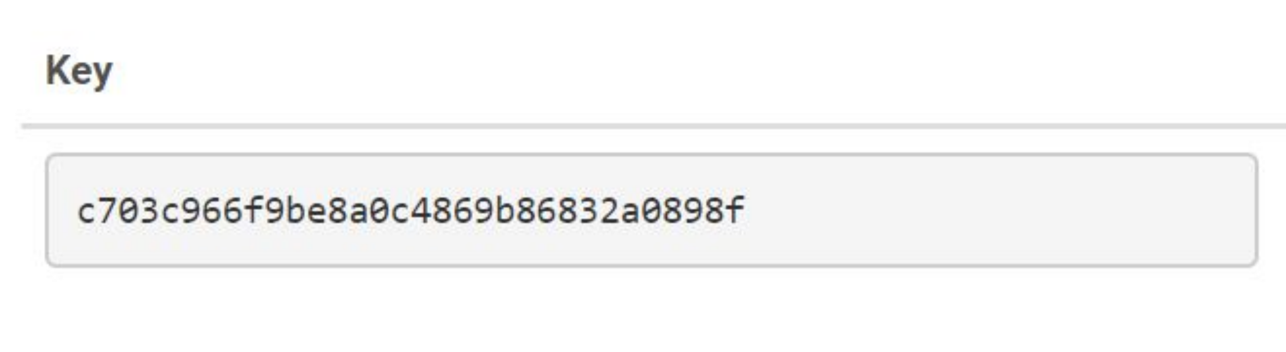
What is an API Key?

❑ API Keys helps *identify* your app & your requests

❑ Used for *authorization* and *authentication*

How do I get one?

❑ You need to register & sign up for an API Key



Key

```
c703c966f9be8a0c4869b86832a0898f
```

Three Parts of API Requests

1. **API Key -** This key acts as a form of authentication, which can lead to access control

2. **Base URL-** This is the URL that you can make a request to the website's API with.

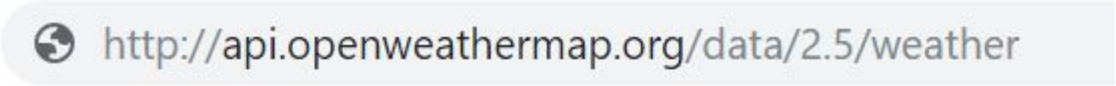
3. **Search Query -** This is the query that is used to get back any information of a particular API

Complete Query = Base URL + API Key + Search Query

Example of API Request

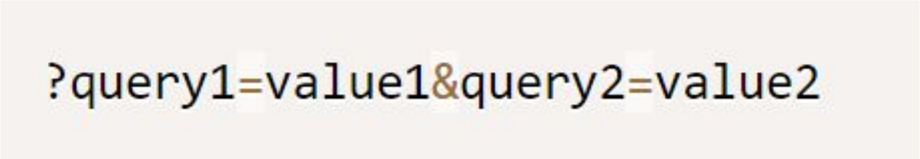
- Start with the **base_url** and add in the query parameters
- Queries always begin with a question mark (?)
- Additional queries are followed by an ampersand (&)

Base Url



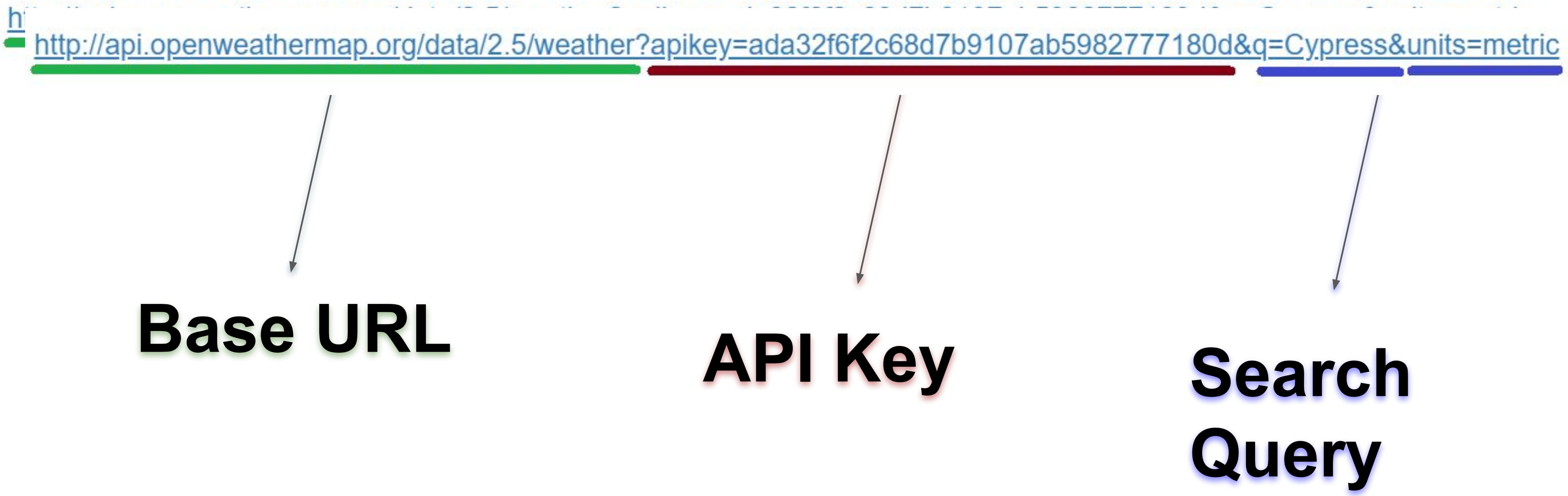
http://api.openweathermap.org/data/2.5/weather

Query Parameters



```
?query1=value1&query2=value2
```

Example of API Request



http://api.openweathermap.org/data/2.5/weather?apikey=ada32f6f2c68d7b9107ab5982777180d&q=Cypress&units=metric

Base URL

API Key

Search Query

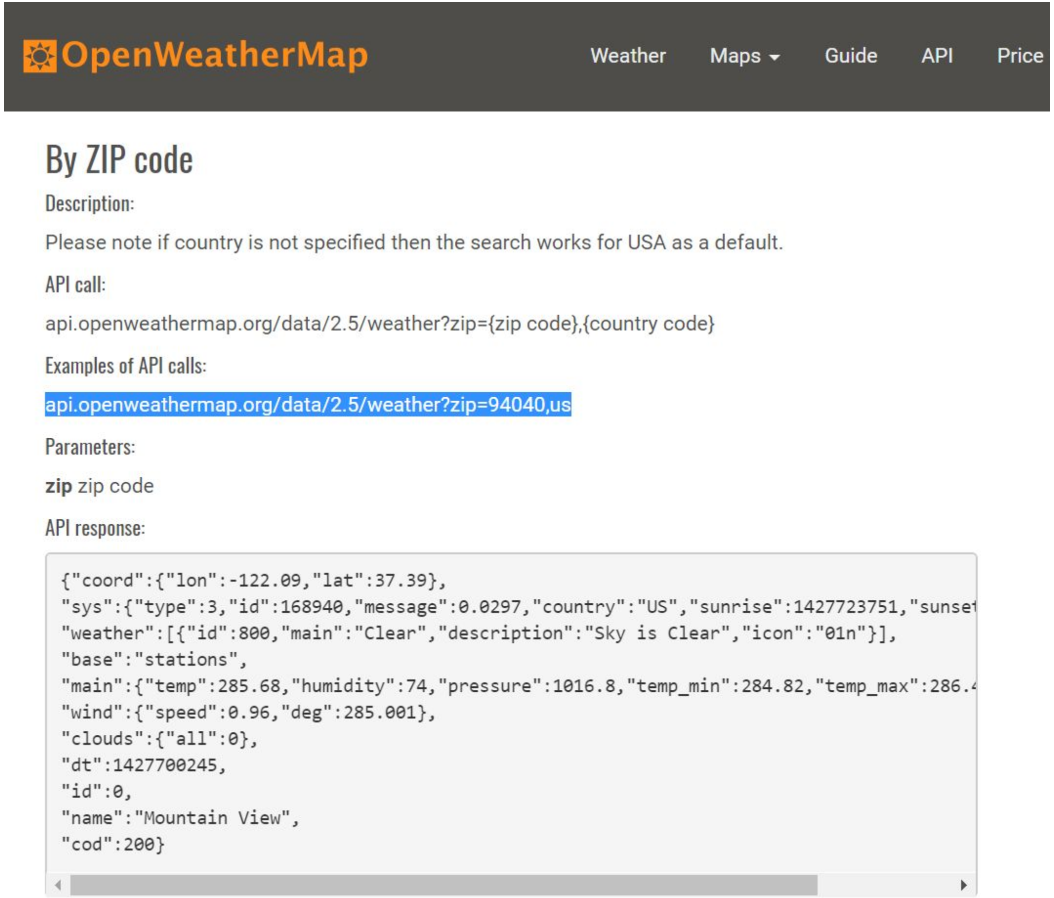
Question: How many query parameter(s) do we have here?

Documentation

OpenWeatherMap Example:

Each Web API will have its own

Documentation for you to follow.



By ZIP code

Description:

Please note if country is not specified then the search works for USA as a default.

API call:

api.openweathermap.org/data/2.5/weather?zip={zip code},{country code}

Examples of API calls:

api.openweathermap.org/data/2.5/weather?zip=94040,us

Parameters:

zip zip code

API response:

```
{"coord":{"lon":-122.09,"lat":37.39},
"sys":{"type":3,"id":168940,"message":0.0297,"country":"US","sunrise":1427723751,"sunse
"weather":[{"id":800,"main":"Clear","description":"Sky is Clear","icon":"01n"}],
"base":"stations",
"main":{"temp":285.68,"humidity":74,"pressure":1016.8,"temp_min":284.82,"temp_max":286.
"wind":{"speed":0.96,"deg":285.001},
"clouds":{"all":0},
"dt":1427700245,
"id":0,
"name":"Mountain View",
"cod":200}
```

Documentation
DEMO

Run this code:

Example of API Call:

http://api.openweathermap.org/data/2.5/weather?zip=94040,us

Does it work?

Documentation
DEMO

REMEMBER:

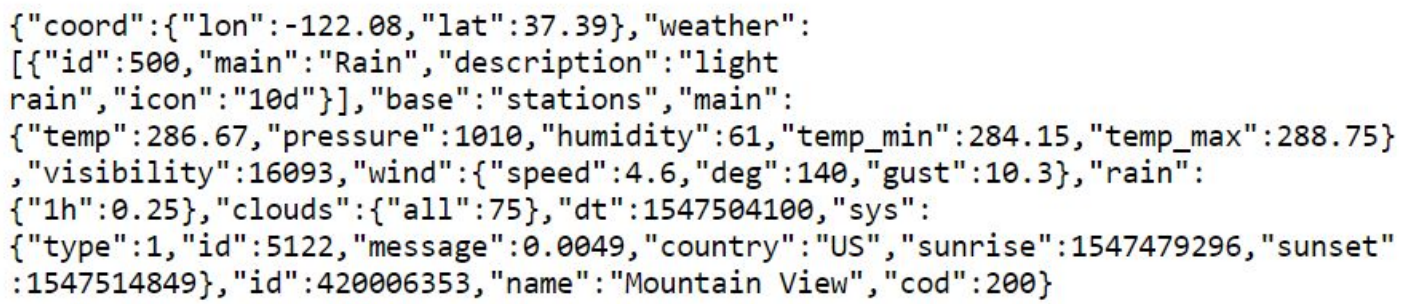
You need to have another query for your **api_key**

Example of API Call:

http://api.openweathermap.org/data/2.5/weather?zip=94040,us&apikey=c703c966f9be8a0c4869b86832a0898f

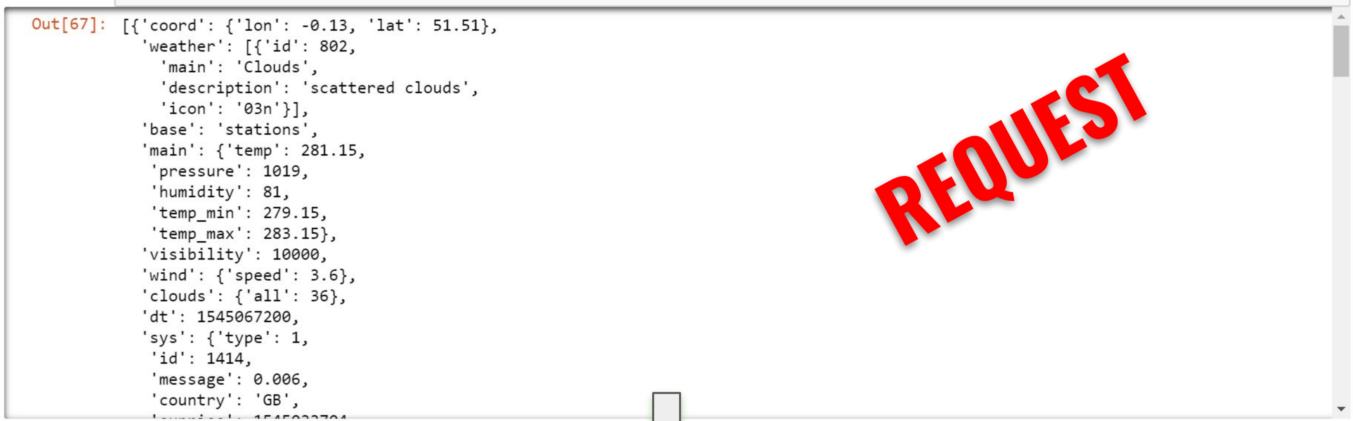
Documentation DEMO

TA-DA!

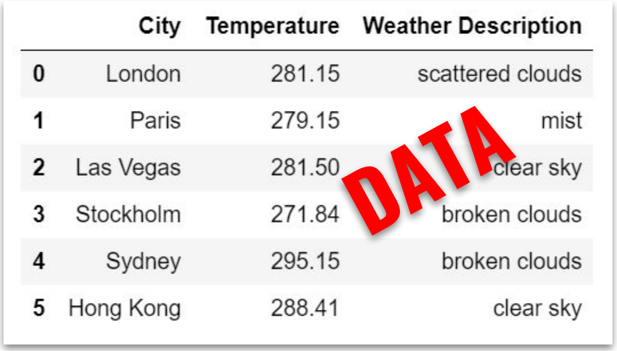


```
{"coord":{"lon":-122.08,"lat":37.39},"weather":
[{"id":500,"main":"Rain","description":"light
rain","icon":"10d"}],"base":"stations","main":
{"temp":286.67,"pressure":1010,"humidity":61,"temp_min":284.15,"temp_max":288.75}
,"visibility":16093,"wind":{"speed":4.6,"deg":140,"gust":10.3},"rain":
{"1h":0.25},"clouds":{"all":75},"dt":1547504100,"sys":
{"type":1,"id":5122,"message":0.0049,"country":"US","sunrise":1547479296,"sunset"
:1547514849},"id":420006353,"name":"Mountain View","cod":200}
```

API Request → Data



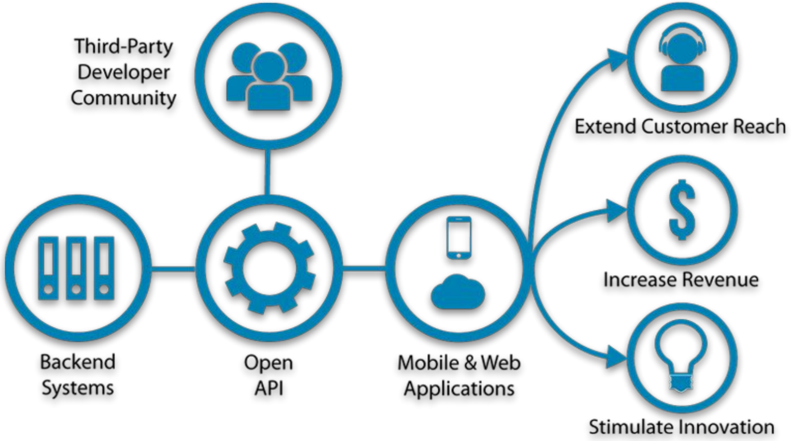
```
Out[67]: [{'coord': {'lon': -0.13, 'lat': 51.51},
           'weather': [{'id': 802,
             'main': 'Clouds',
             'description': 'scattered clouds',
             'icon': '03n'}],
           'base': 'stations',
           'main': {'temp': 281.15,
            'pressure': 1019,
            'humidity': 81,
            'temp_min': 279.15,
            'temp_max': 283.15},
           'visibility': 10000,
           'wind': {'speed': 3.6},
           'clouds': {'all': 36},
           'dt': 1545067200,
           'sys': {'type': 1,
            'id': 1414,
            'message': 0.006,
            'country': 'GB',
```



	City	Temperature	Weather Description
0	London	281.15	scattered clouds
1	Paris	279.15	mist
2	Las Vegas	281.50	clear sky
3	Stockholm	271.84	broken clouds
4	Sydney	295.15	broken clouds
5	Hong Kong	288.41	clear sky

REST API Summary

- ❑ **API - Application Programming Interface**
- ❑ It’s when two applications *talk* together
- ❑ APIs simply provide data
- ❑ API's are part of a company's server that receives requests and sends responses


Third-Party Developer Community
Backend Systems
Open API
Mobile & Web Applications
Extend Customer Reach
Increase Revenue
Stimulate Innovation