When a client uses our company's website to search for hotels, our search engine will gather information from a variety of websites based on the client's preferences through **APIs**. An API call is very similar to navigating to a website. An API points to a URL and collects some data from the webpage or server.

When clients request information from our server through our website, they are making an API call. Once our database has the client's search criteria, our servers search the web for hotels on behalf of the client. Now the roles are reversed: our company is the client requesting information, and all the websites where we derive information are the servers

sing an API has its limitations because not all information from a server is accessible. Most APIs have tiered services, from free to paid. Free services allow access to limited information, and paid subscriptions provide more access based on the payment plan. Our company has a paid subscription for APIs, but we can only get certain information from websites on hotels such as location, accessibility, rooms, prices, services, and amenities, as well as regional weather data.

Now that you have a general concept of how APIs work, let's register for an OpenWeatherMap **API key**, a token granting access, and use it to retrieve weather data

**The JavaScript Object Notation Format for API Data**

The API has reached the website or server, its endpoint, and now we can retrieve data from the website. When we retrieve data from a website, we have to make a "request," which returns data in a text format, not in a tab- or comma-separated file.

One format we can use to parse data is **JavaScript Object Notation (JSON)**. The JSON format is also referred to as an "object" or "JSON object." The data inside a JSON object opens and closes with curly braces, much like a Python dictionary. Inside the JSON object is a collection of dictionaries and arrays.

Below is an example of what weather data looks like in the JSON format when we request it from the OpenWeatherMap website. There are curly braces that wrap the data, and inside the curly braces are dictionaries and arrays.

## The Python Requests Library

To request JSON data over the internet, we use the Requests Library in Python.

1. Launch the Anaconda Prompt for your PythonData environment.
2. After the Python prompt, >, type python to launch Python.
3. At the Python prompt, >>>, type import requests and press Enter.
4. On the next line type requests.\_\_version\_\_ and press Enter.
5. The output should be version 2.22.0 or later.

My version is: '2.27.1'