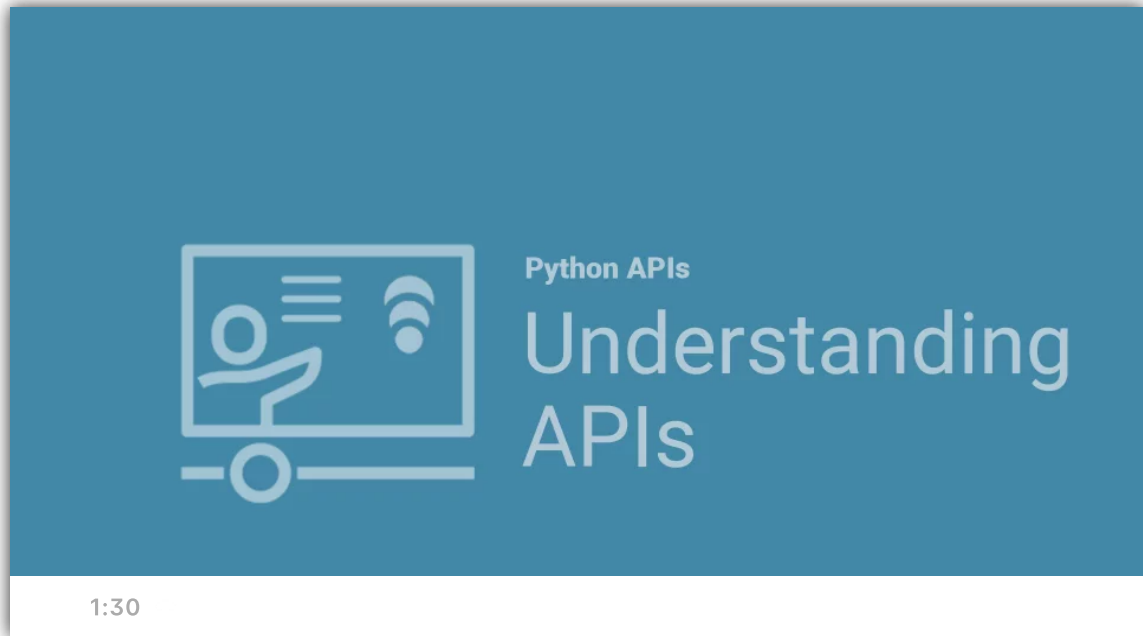


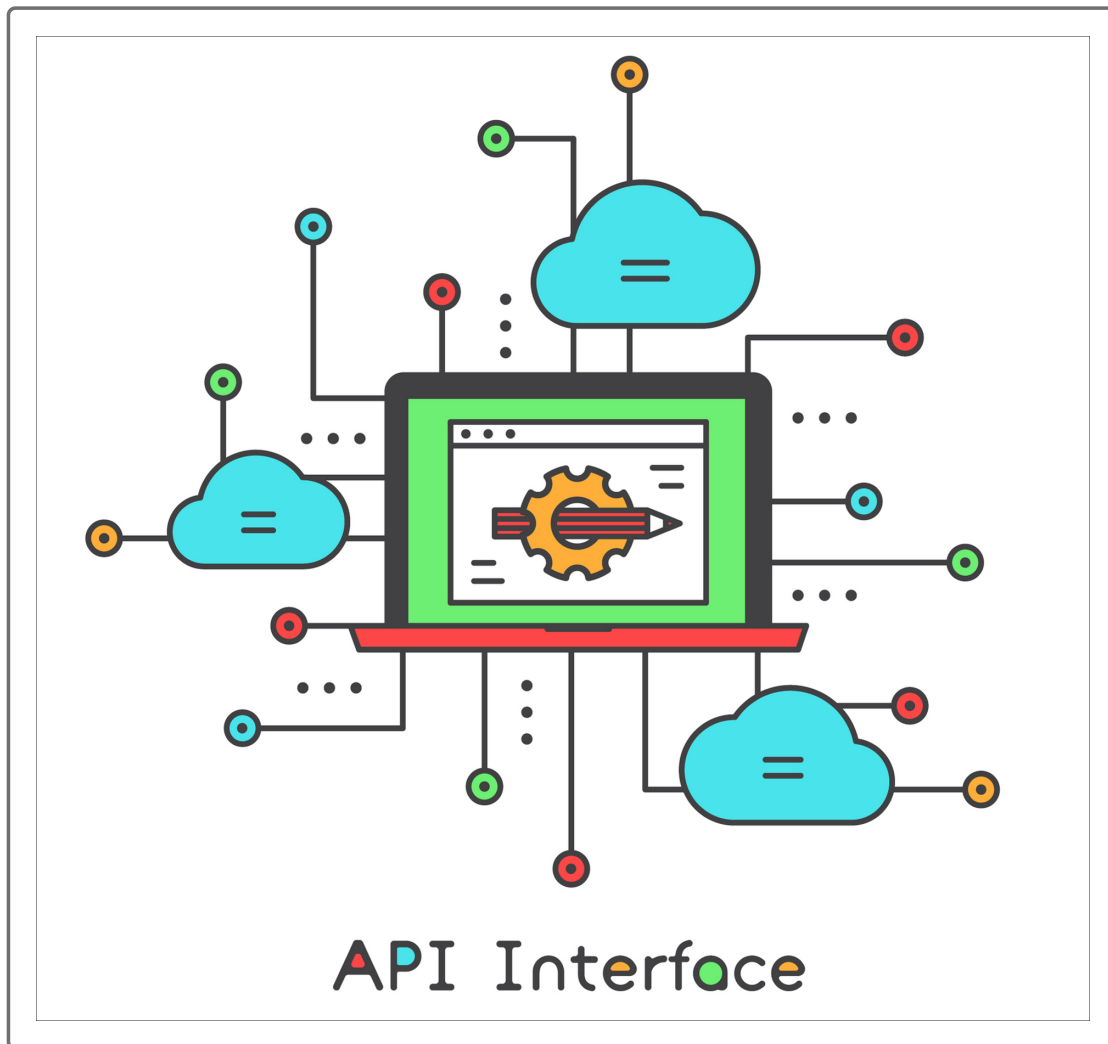
6.2.1 Understanding APIs

Time to get some weather data. During a brainstorming session with Jack, you decide to use the OpenWeatherMap Application Programming Interface (API) to get the weather data for your database. Of course, now comes the tricky part: actually getting the data. You know you can do this using an API—but how do you use an API? Time to dig back in. The weather data is visually displayed on maps for the customer. But, in order for that to happen, you'll need to retrieve the weather from each city in your database. Since this is your first time using an API to get data from a server, your manager would like you to review APIs and how your company retrieves the data.



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.



Using 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.