Welcome to Application Program Interfaces, or APIs.

After watching this video, you will be able to define an API, list API libraries, and define REST API in relation to request and response.

An Application Programming Interface, or API, allows communication between two pieces of software.

For example, in a program, you have some data and other software components.

You use the API to communicate using inputs and outputs without knowing what happens at the back end.

The API only refers to the interface.

It is the part of the library you see while it contains all the program components.

Now, to further understand how an API works in a library, consider an example of the Pandas library.

Pandas is a set of software components where not all components are written in Python.

In your program, there is some data and a set of software components.

You can use the Pandas API to process the data by communicating with the other software components.

The software component at the back end can be the same, but there can be an API for different languages.

Consider TensorFlow at the back end, written in C++, that can use APIs for other languages, such as Python, JavaScript, C++, Java, and Go, and thus, the API is just the interface.

Other volunteer-developed APIs for TensorFlow are Julia, MATLAB, R, Scala, and many more.

So, REST APIs are another popular type of API.

The R-E stands for Representational, the S stands for State, and the T stands for Transfer.

They allow you to communicate through the internet and take advantage of resources, like storage, data, artificially intelligent algorithms, and much more.

In REST API, your program is the client.

The API communicates with a web service you can call through the internet, though there are rules regarding communication, input or request, and output or response.

So, let's look at some common terms used with regards to API.

You or your code are the client.

The web service is the resource.

The client finds the service via an endpoint.

And the client sends requests to the resource and receives a response from the resource.

Data is transmitted over the internet using HTTP methods.

The REST APIs get all the information from the request sent by the client.

The request is sent using an HTTP message that contains a JSON file.

The file contains instructions for what operation is to be performed by the web service.

This operation is transmitted to the web service via the internet, and the service performs the operation.

Similarly, the web service returns a response through an HTTP message where the information is returned using a JSON file.

And this information is transmitted back to the client.

Now, another example of a REST API is Watson Speech-to-Text API.

This API converts speech to text.

In the API call, you will send a copy of the audio file to the API.

This is called a post request.

In the API, we'll send the text transcription of what the individual is saying.

At the back end, the API is making a GET request.

And finally, let's look at our final example, the Watson Language Translator API.

You send the text you would like to translate into Watson Language Translator API.

The API will translate the text and send the translation back to you.

And in this case, the API translates English to Spanish.

In this video, you learned an Application Programming Interface, or API, allows communication between two pieces of software.

An API is the part of the library you see while the library contains all the components of the program.

And REST APIs allow you to communicate through the internet and take advantage of resources like storage, data, artificially intelligent algorithms, and much more.