# [Capstone Project (Module 1)](https://www.coursera.org/learn/ibm-data-analyst-capstone-project/lecture/v5WIG/course-introduction)

Table of Contents

[Capstone Project (Module 1) 1](#_Toc188282677)

[Collecting Data Using APIs 1](#_Toc188282678)

[Lab 1: Review of Accessing APIs 1](#_Toc188282679)

## Collecting Data Using APIs

### Lab 1: Review of Accessing APIs

When you access a web page, your browser send an HTTP request to the server hosting the page. The server attempts to find the requested resources, typically starting with “index.html”. if found, the server responds with the resource, providing details like type, length, and additional information.

The process involves the client (your browser) and the web server, which stores resources like HTML files, images, and text files. The HTTP protocol facilitates sending and receiving web data, including web pages and images. In this lab, you’ll use the Requests library to interact with HTTP.

A diagram of a response

Description automatically generated

**Uniform Resource Locator (URL)**

Uniform resource locator (URL) is the common way to find resources on the web. A URL can be broken down into three main parts:

* **Scheme**: The protocol used, which in this lab will always be http://
* **Internet address or Base URL**: Used to locate the resource. Examples include www.ibm.com and  www.gitlab.com
* **Route**: Location on the web server for example: /images/IDSNlogo.png

You may also hear the term Uniform Resource Identifier (URI). URL are actually a subset of URIs. Another popular term is endpoint, which refers to the URL of an operation provided by a web server.

**Request**

The process can be broken into the **Request** and **Response**process. The request using the get method is partially illustrated below. In the start line we have the GET method, this is an HTTP method. Also the location of the resource /index.html and the HTTP version. The Request header passes additional information with an HTTP request:

**A white box with black text

Description automatically generated**

When an HTTP request is made, an HTTP method is sent, this tells the server what action to perform. A list of several HTTP methods is shown below. We will review more examples later.

A white text with black text

Description automatically generated

**Response**

The figure below represents the response; the response start line contains the version number HTTP/1.0, a status code (200) meaning success, followed by a descriptive phrase (OK). The response header contains useful information. Finally, we have the response body containing the requested file, an  HTML  document. It should be noted that some requests have headers.

**A close-up of a response message

Description automatically generated**

*A yellow and black text on a white background

Description automatically generated*

*Types of Response Message*

**Get Request with URL Parameters**

You can use the **GET** method to modify the results of your query, for example, retrieving data from an API. We send a **GET** request to the server. As before, we have the **Base URL**, in the **Route** we append /get, this indicates we would like to perform a GET request. This is demonstrated in the following table:

**A white rectangular box with black text

Description automatically generated**

A [query string](https://en.wikipedia.org/wiki/Query_string?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkPY0101ENSkillsNetwork19487395-2021-01-01) is a part of a uniform resource locator (URL), it sends other information to the web server. The start of the query is a ?, followed by a series of parameter and value pairs, as shown in the table below. The first parameter name is name and the value is Joseph. The second parameter name is ID and the Value is 123. Each pair, parameter, and value is separated by an equals sign, =. The series of pairs is separated by the ampersand &.

**A white rectangular box with colorful text

Description automatically generated with medium confidence**

**Post Requests**

Like a GET request, a POST is used to send data to a server, but the POST request sends the data in a request body. In order to send the Post Request in Python, in the URL you can change the route to POST:

### Lab 2: Collecting Data Using APIs

See ../Scripts/Collecting\_job\_data\_using\_APIs-Lab.ipynb

## Collecting Data Using Web Scraping