6.1 Lesson Summary - APIs

The most important component to Data Analytics is the data itself. The best models and analysis are meaningless without high-quality data to analyze. Over the past 30 years the Internet has revolutionized the way data is shared. The most familiar benefit of the Internet are websites and apps that let us share and consume. We might use the Internet to catch up on a days headlines or use a streaming service to listen to music. The same mechanisms that allow the sharing of website and app information are ideal for sharing other kinds of data. Using Python we can access and begin to analyze this data.

Concept: **Server-Client** architecture is the most common way information is exchanged across networks and the Internet. In a server-client model the server is an application or device that has been set up to share information with other applications or devices. The client in the server-client model is an application or machine that accesses information from the server. For example, if you are watching a video on your phone your phone is the client and the computer sending you the video information would be the server.

Concept: An Application Programming Interface or **API** is an established set of interactions by which different applications can interact with the application hosting the API. When data is shared over the web it is usually shared using a web API. Web APIs vary some from site to site but generally once you are comfortable accessing data from one API you can reuse the same techniques to access data from any web API.

Concept: Data from web APIs can be structured in a number of different ways. One of the most common is by using JavaScript Object Notation or **JSON**. JSON allows you to use text to organize data by objects. For example if you wanted to describe all of the customers in database a JSON representation of this could look like the following:

{[

"customer1":{

"name": "Susan",

"purchases": 7

},

"customer2":{

"name": "Joe",

"purchases": 12

}

]}

Concept: You can use Python's *requests* library to access data from web API's. The *get* method in the *requests* library performs an HTTP get request which is the standard way to retrieve data from a web API. To accomplish this, you could use the following code:

*import requests*

*request\_response = requests.get(request\_url)*

* Activity: 01-Ins\_RequestsIntro, 02-Stu\_SpaceX

Concept: After a response object has been generated from your get request, parsing the data is easier if you create a dictionary object based on the JSON data retrieved from the web API. You can then access your data just as you would from any other dictionary object. To create a dictionary from your get request you could use the following code:

*response = requests.get(url)*

*response\_json\_dict = response.json()*

To print the contents of a dictionary in a JSON format you can use the *dumps* method from the *json* library. For example:

*import json*

*json.dumps(response\_json\_dict, indent=4, sort\_keys=True)*

* Activity: 01-Ins\_RequestsIntro, 02-Stu\_SpaceX, 03-Ins\_ManipulatingResponses, 04-Stu\_FarFarAway

Concept: It is often necessary to perform multiple HTTP requests to retrieve all of the necessary data. You can perform multiple requests and combine the results into a response list. For example:

responses = []

for url in url\_list:

response = requests.get(url)

responses.append(response.json())

* Activity: 09-Ins\_IterativeRequests