Name: Deepika Mahajan

Neptune Code:T1VISF

Team: 4

Project : FOREX Trading System

Task: Data Aquisition from Open source Finance Data API.

API used: Quandl and Twitter API.

1. **Quandl:Financial Data API**

What is Quandl used for?

The world's most powerful data lives on **Quandl**.The premier source for financial, economic, and alternative datasets, serving investment professionals. **Quandl's** platform is used by over 400,000 people, including analysts from the world's top **hedge funds**, **asset managers** and **investment banks**.

Feature of Quandl: Simple, Powerful And Free

Your Choice of Data is very important when it comes traiding in **Stock** market.

Who are Quandl?

Founded in 2013, Quandl has become a respected data provider. They now boast over 250,000 users from individuals to large hedge funds and investment banks. **Quandl provides a wide ranging dataset across two main categories:**

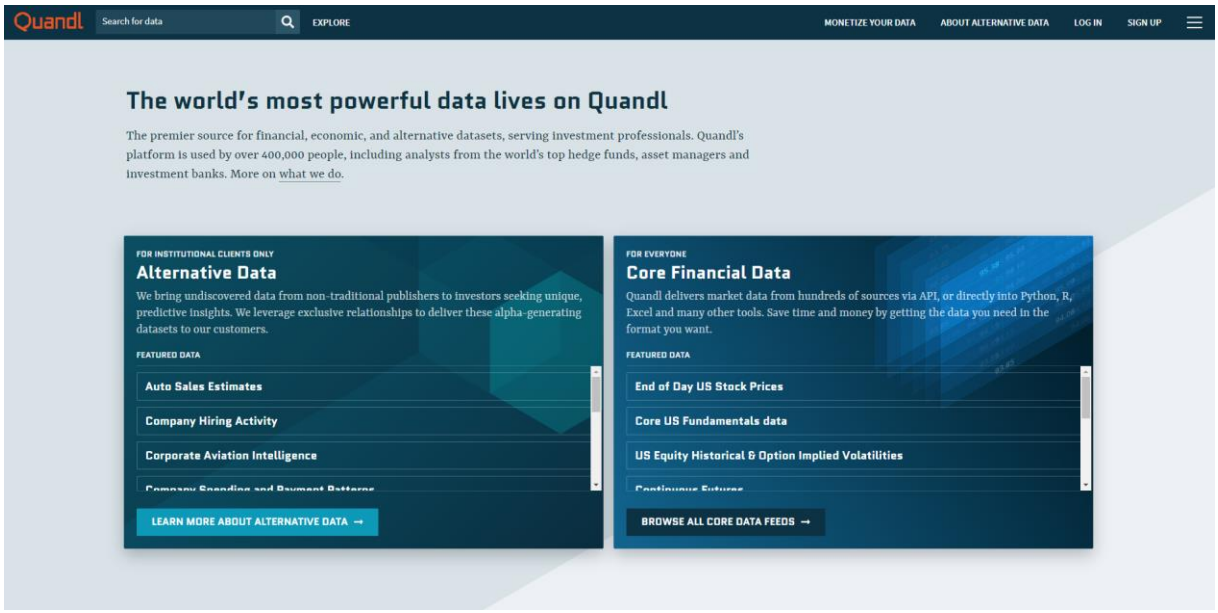
- **Core Financial Data:** coverage of securities and market data across all asset classes.
- **Alternative Data:** a vast collection of data sources from outside of the normal or regulatory data companies or markets.

Quandl is an aggregated marketplace for financial, economic and other related APIs. Quandl aggregates APIs from third-party marketplaces as services for users to purchase whatever APIs they want to use.

How Quandl helps you invest & trade?

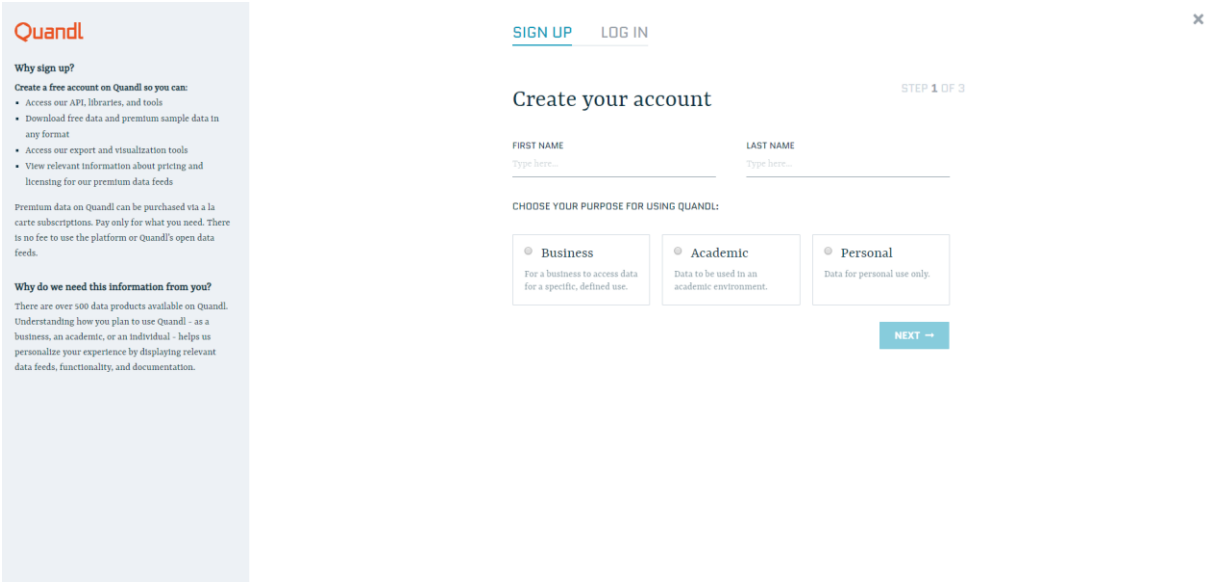
Quandl provides a wide range of data and importantly a lot of flexibility in how you consume that data. Client libraries for data access in Ruby, R, Excel and **Python** are available, as well as an open and well documented API that can be integrated with any language of your choice.

How Quandl's Excel and Python clients can be used to obtain data?



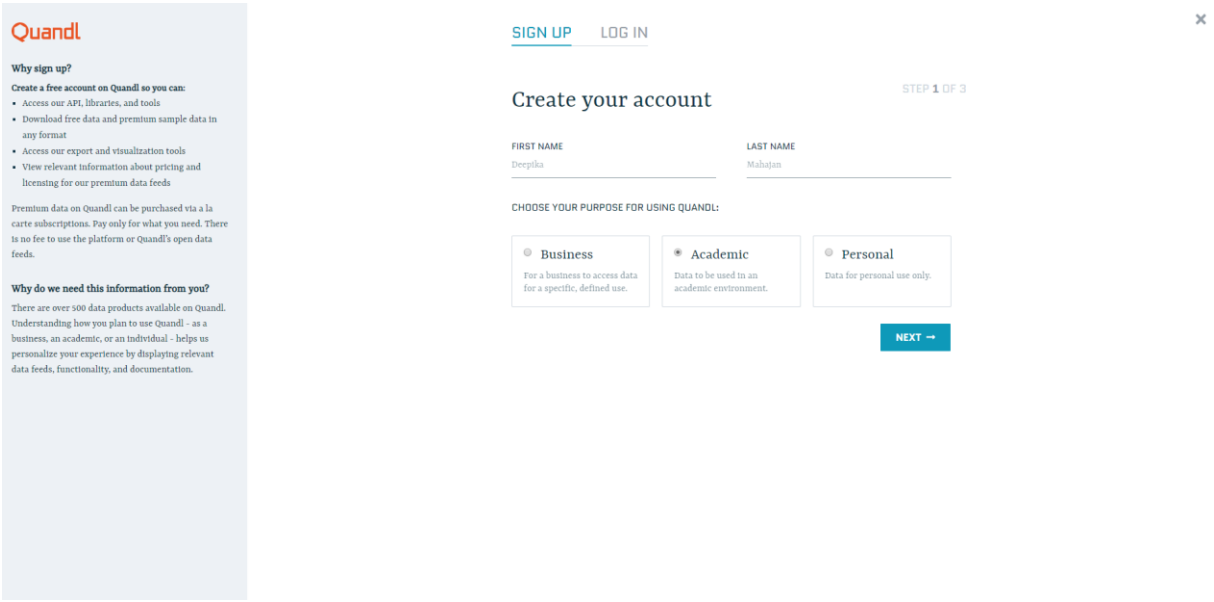
Step 1: Quandl Signup:

- First step is to signup here:

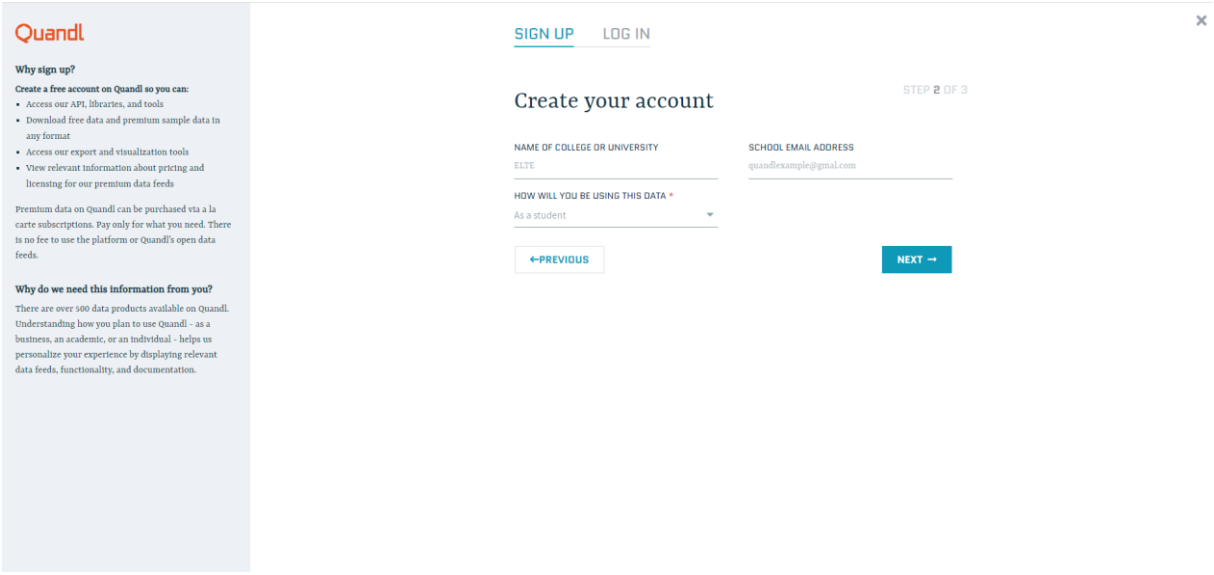


- Enter your email address

- Select a reason for using the data
- Select the purpose of using Quandl.

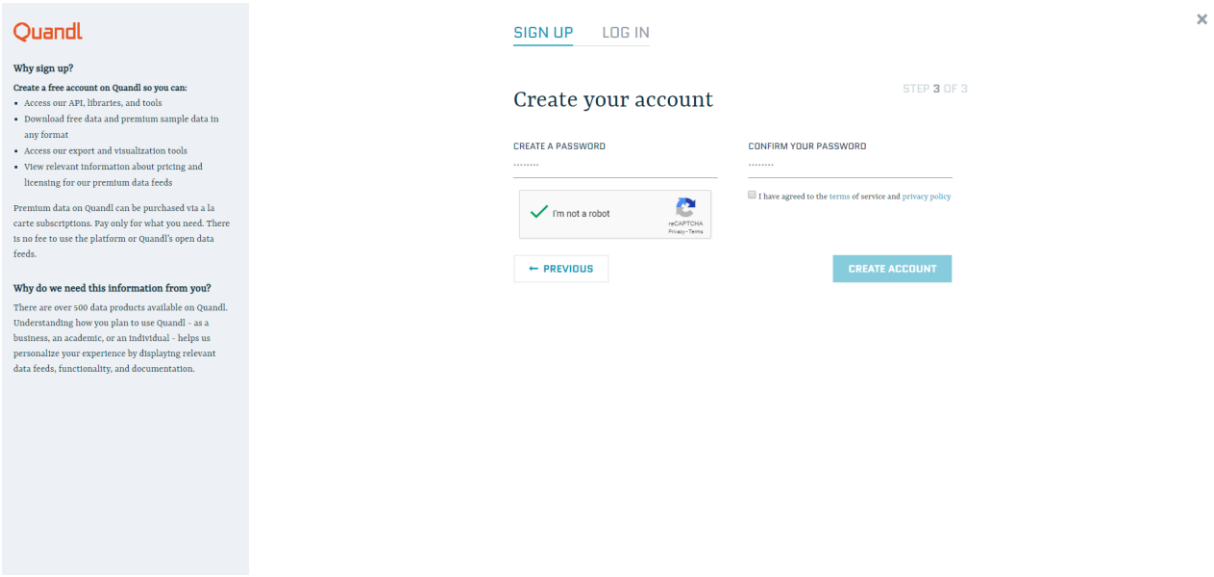


- Enter your University and School address:



Create Password

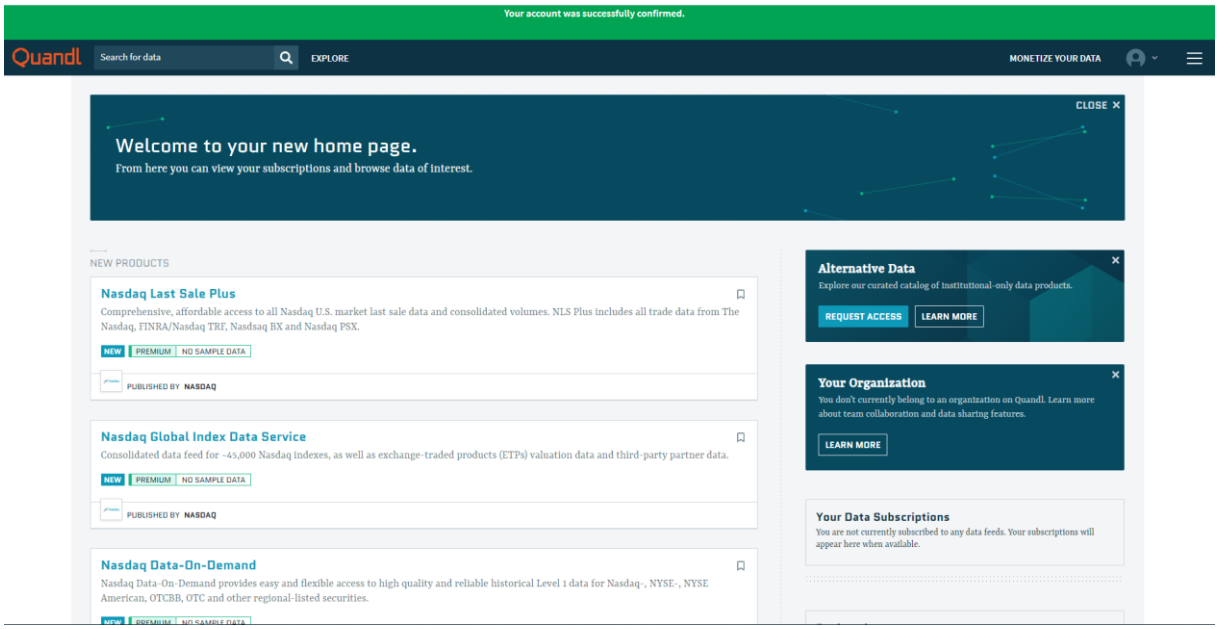
- Enter a secure password
- Press ‘Create Account’



- You will receive an email verification that you must select to activate the account.

After login we can access all data free and paid. Below is the Dashboad of Quandl:

Quandl Dashboard:



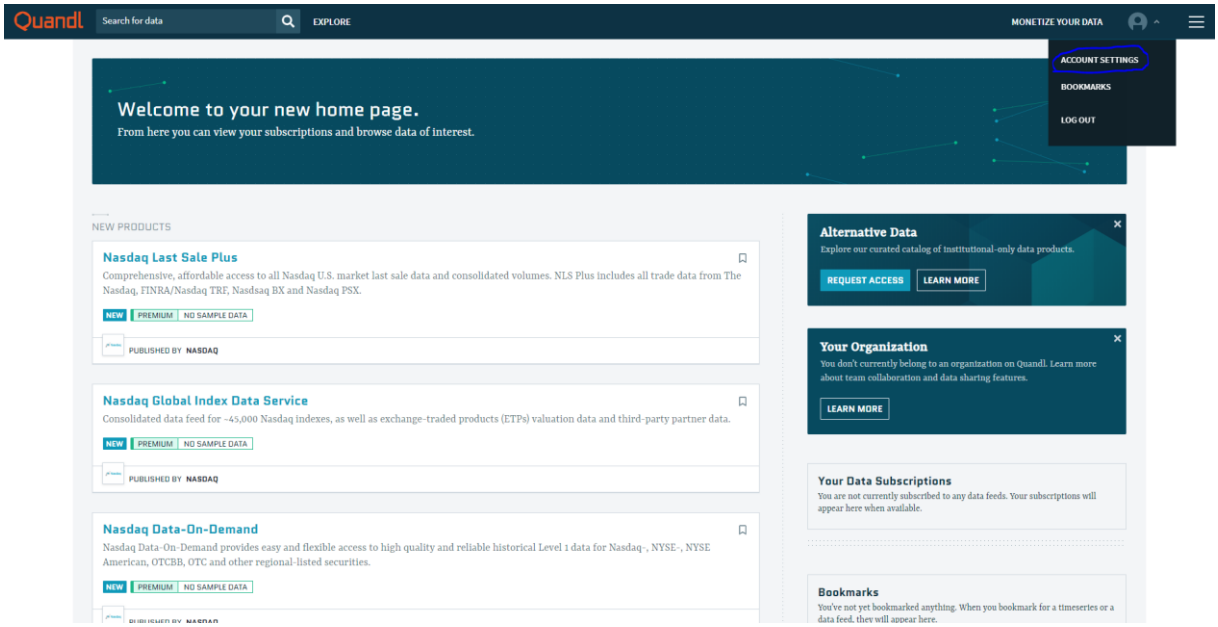
Quandl APIs:

After loging to Quandl he first step in that process is to **Generate the appropriate API key**.

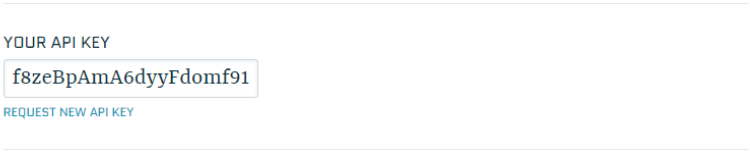
Quandl API Key:

To work with a **Quandl APIs** you must first ensure that you have a **Quandl API client key**. **API Client key** is used when you want to **access data from Quandl**.Go to the Account Settings page in your Quandl account:

Select Account Setteng in the top right corner. **(Marked in a blue circle).**



API key which is a long string of random characters is displayed.



Further I am going to use 2 methods to gather data:

1. Excel API

2. Python API

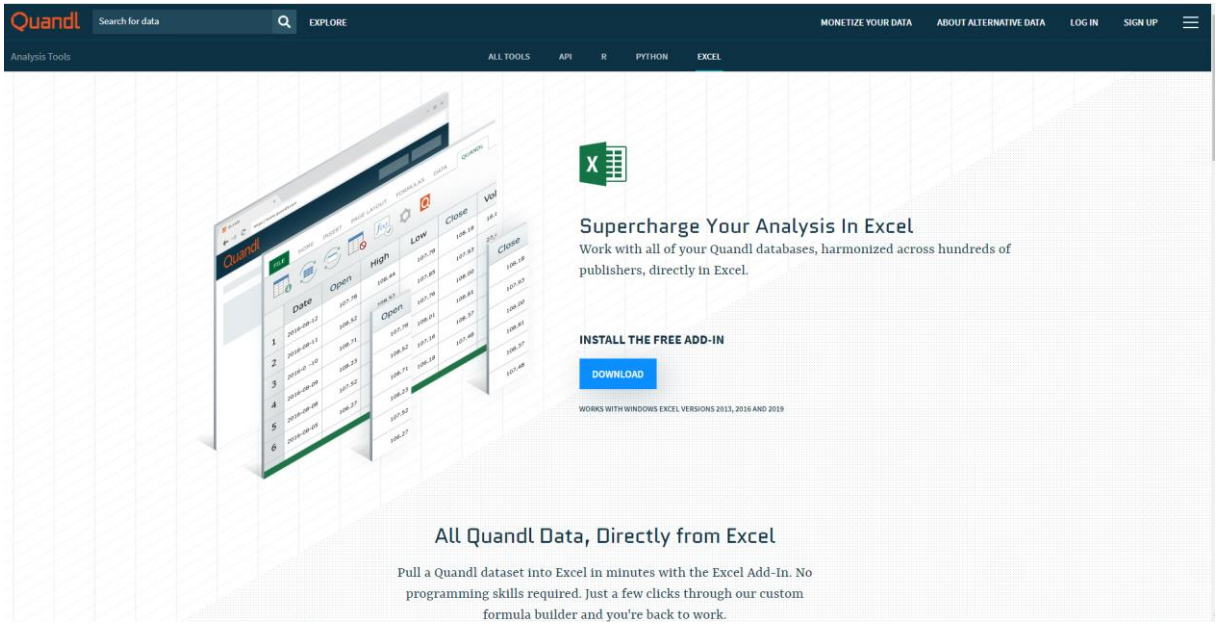
Quandl makes data acquisition easy. There are many menthods and ways to gather data unsing quandl. I am focusing on Python and Excel methods to gather data efficiently.

Excel API:

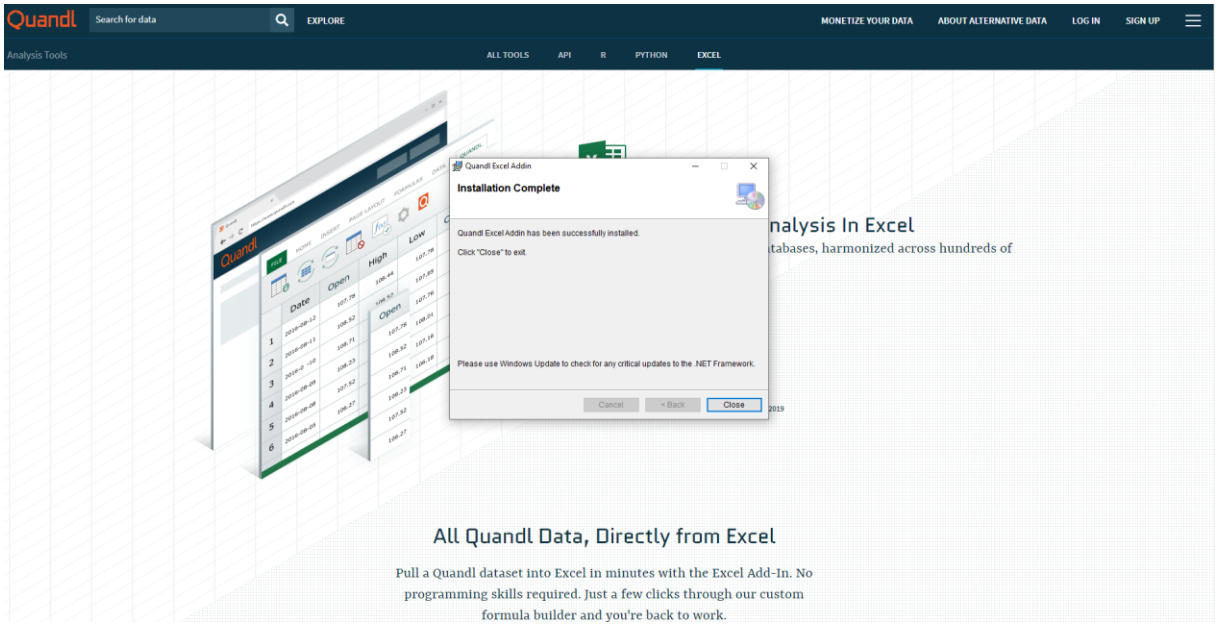
Quandl provides an excellent Excel Add-on which automatically integrates with any supported version of Windows Excel. Currently supported Excel versions are 2010, 2013 & 2016.

Quandl Excel Add-On Configuration: Pulling all data into Excel

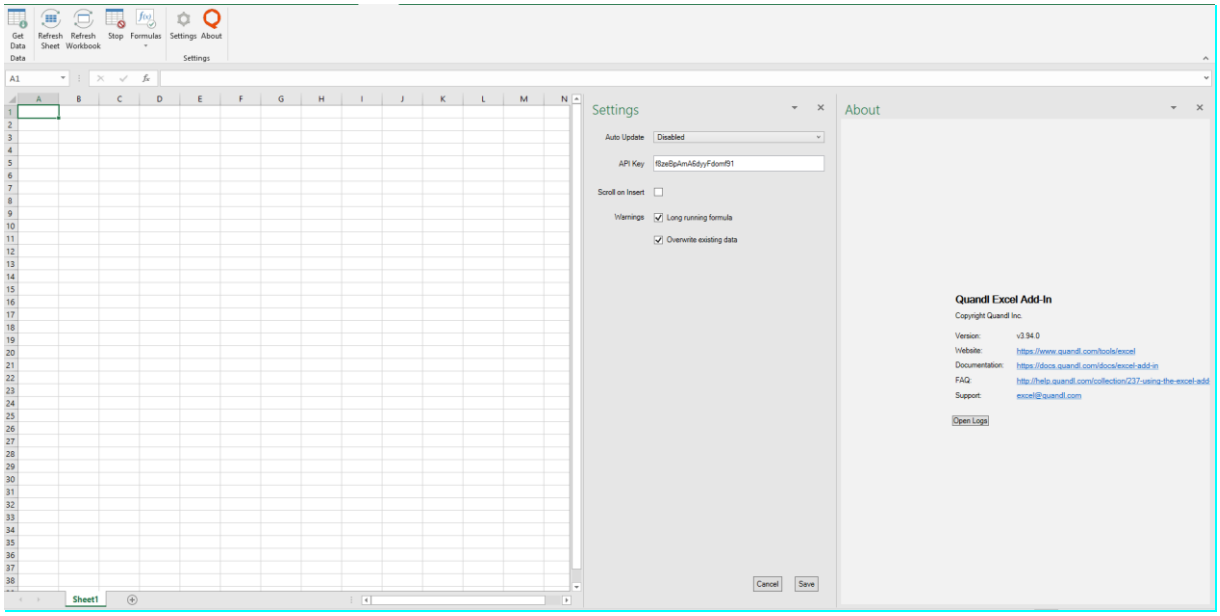
The add on is available at https://www.quandl.com/tools/excel. To install run the downloaded installer file and follow the instructions. After installation the next time you open Excel you will be asked permission to install the Excel customisation:



Download this Add-In and install.

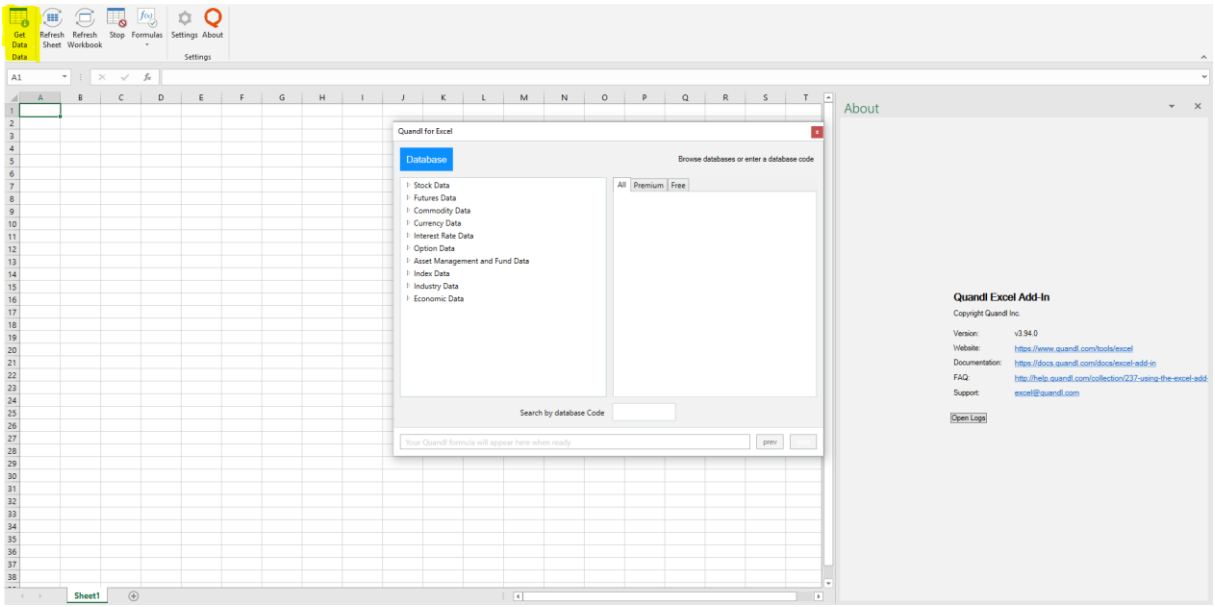


Now Open Excel a new blank worksheet and you should see a **'Quandl'** menu item in Excel. Select this menu item and enter the **API Key** to complete the configuration:



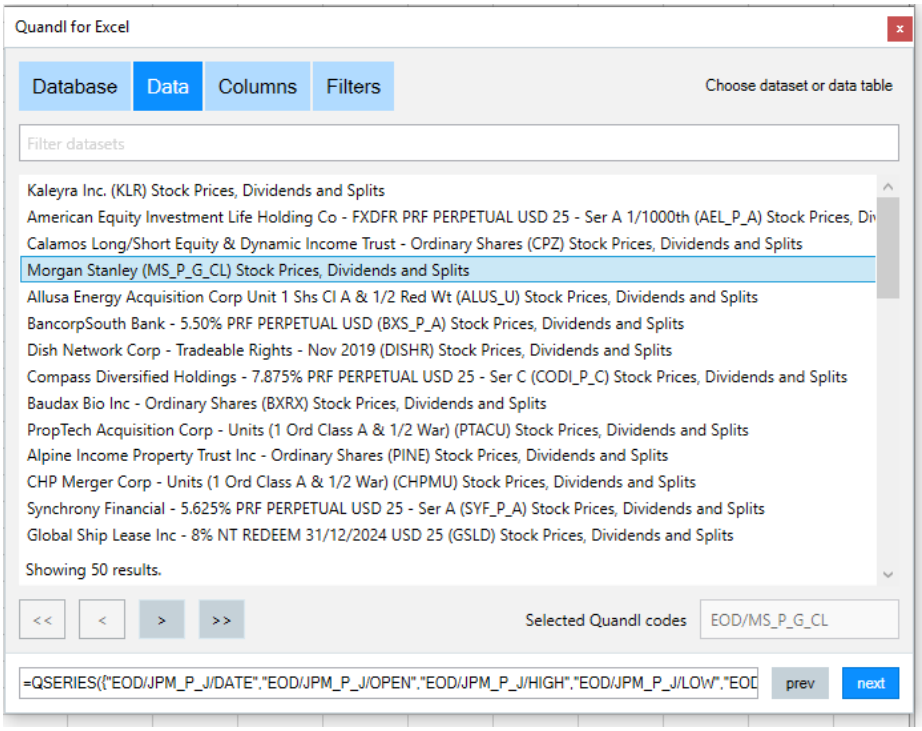
Using the Formula Builder to obtain historic data

1. Open the Quandl Formula Builder within Excel:

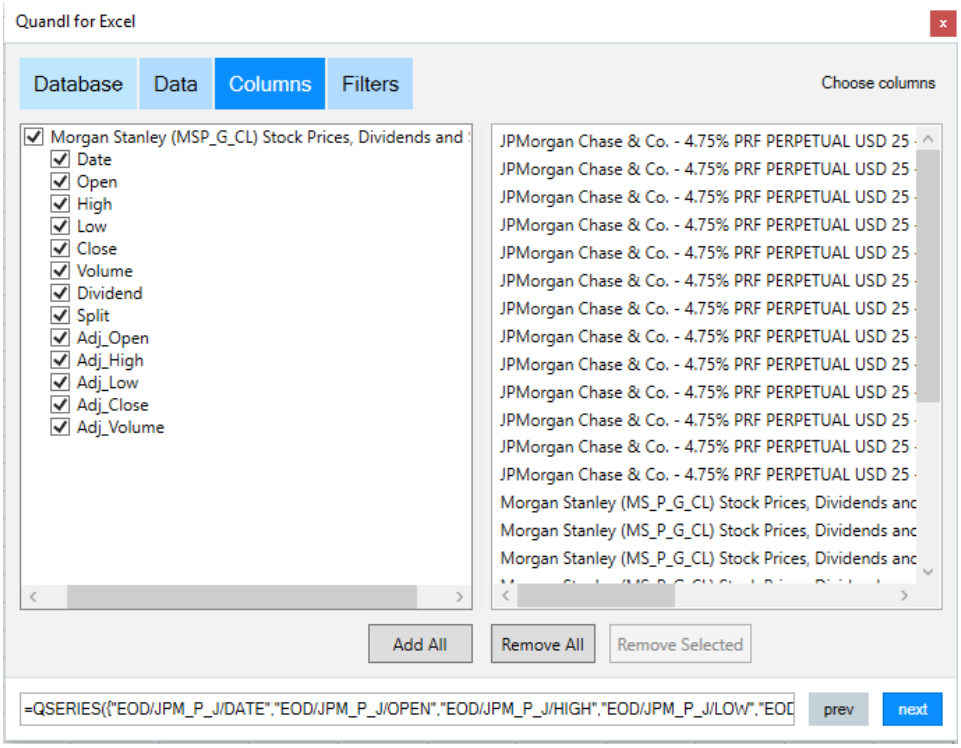


Click on Get Data. Now you can choose any of the data from Quandl. And import it for use.

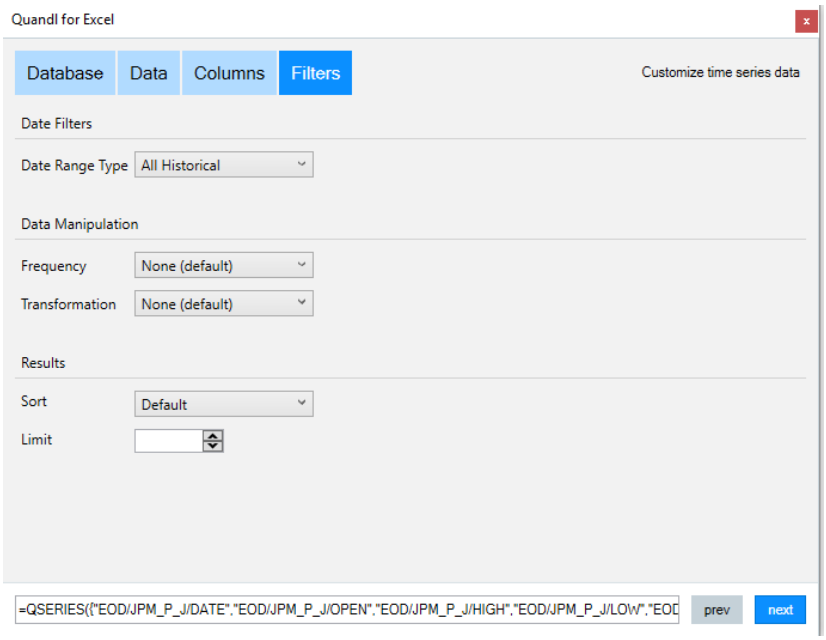
2. Select a data set to download: I choose **JP Morgan data**
3. Choose the data points for the time series you want to download:



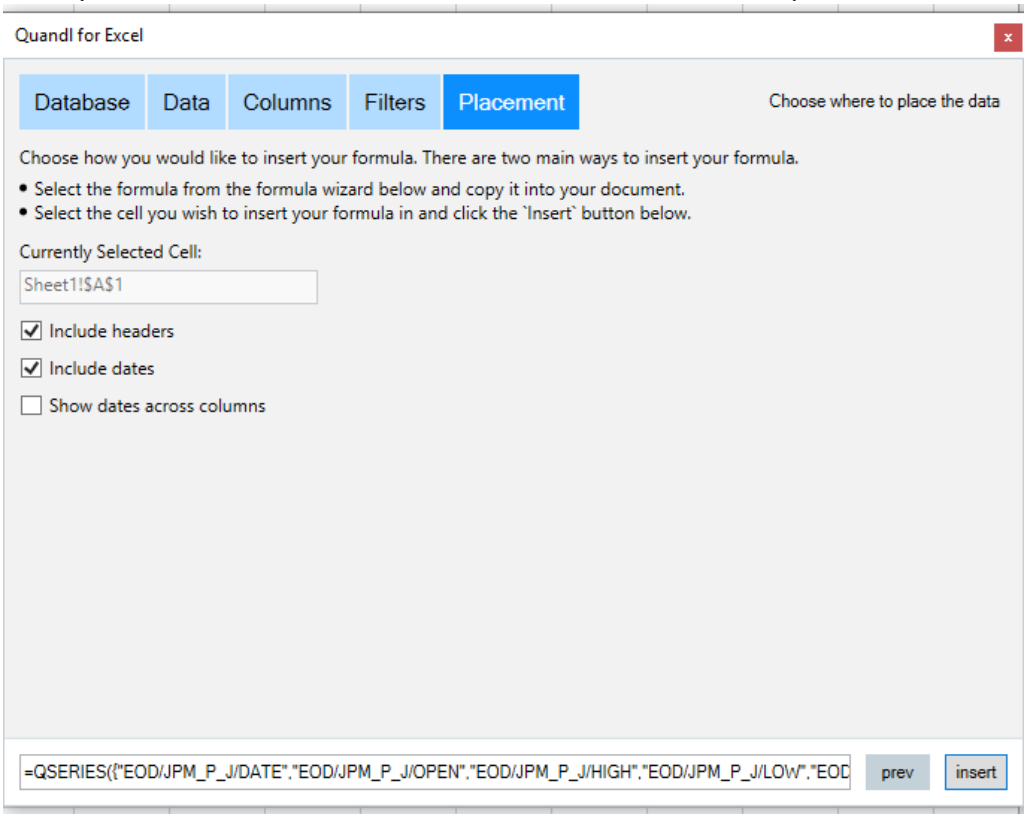
4. Click Next: Then choose the data points for the time series you want to download:



5. In the next tab you have the option to filter the data to be downloaded. You can chooses the date range to pull data for, the frequency of the series (Daily/Weekly/Monthly etc) and sort or limit the data returned to your needs.



6. Finally we confirm the data to be downloaded and where to place it in our Excel file.



7. Data will be available in the Excel sheet. At any time you can refresh the data using the built in tools.

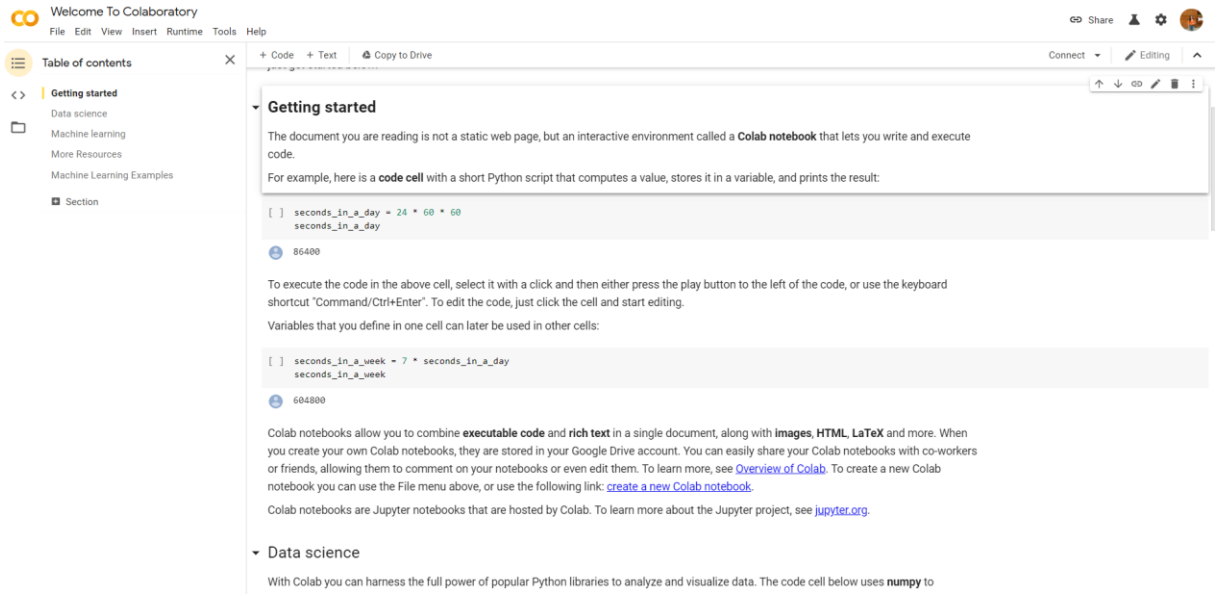
---

2. Python API:

The second approach for data integration we will look at is via the **python** client provided by Quandl.

Setting up your python environment:

For python I am using colab notebook to gather data from Quandl.



- API that is chosen for this task is provided by https://www.quandl.com/
- Specifically, the data related to stock prices, dividends and splits API - https://www.quandl.com/databases/WIKIP
- As I mentioned in the above signing task. This API is freely available upon signing up.
- The API key required for the extration of data is ***qPPFXCRW6twydcQXVB2i***

Here is the link of my work in Python:

I have gathered data from into the csv file.

https://colab.research.google.com/drive/1sc3MGyuhCdbFrD9Xi9bqOuO0mzc2zE4-

Steps:

1.Importing Usefull python libraries

```
!pip install https://github.com/matplotlib/mpl_finance/archive/master.zip
!pip install mpl-finance
import pandas as pd
from pandas import Series
import numpy as np
import requests
import csv
import matplotlib.pyplot as plt
import warnings
from matplotlib import dates
import matplotlib as mlp
from mpl_finance import candlestick_ohlc
%matplotlib inline
```

2. Choose web API

API Selected:

Choosen API for data acquisition provided by https://www.quandl.com/ Specifically, the data related to stock prices, dividends and splits API - https://www.quandl.com/databases/WIKIP

Quandl API is freely available upon signing up.

The API key required for the extration of data is f8zeBpAmA6dyyFdomf91

```
[ ] #Task 2: Retrieve data from the chosen API
    #Function to request data from the url provided using the 'companies' argument and the global variable 'apikey'
    def get_raw_data(companies):
        url = "https://www.quandl.com/api/v3/datatables/WIKI/PRICES.csv?ticker="+companies["Alphabet"]+"&&api_key="+apikey
        get_response = requests.get(url) #Requests to get data from url
        if get_response.status_code != 200: #Retrieval Error Response
            print("Unable to retrive data",get_response.status_code)
        else:
            print("Data Retrieved & Status Code:",get_response.status_code)
            data = csv.reader(get_response.text.strip().split('\n')) #Splits the data on the basis of newline and stores the result
            return data
```

3. Collected raw data, and Storing it in a File

```
apikey = "f8zeBpAmA6dyyFdomf91"        #API KEY retrieved on Signup
```

```
Data Retrieved & Status Code: 200
Raw File Saved
```

4. Load and Represent the data using Pandas DataFrame.

	ticker	date	open	high	low	close	volume	ex-dividend	split_ratio	adj_open	adj_high	adj_low	adj_close	adj_volume
0	GOOG	2018-03-27	1063.00	1064.84	996.92	1005.10	3029471.0	0.0	1.0	1063.00	1064.84	996.92	1005.10	3029471.0
1	GOOG	2018-03-26	1046.00	1055.63	1008.40	1053.21	2558385.0	0.0	1.0	1046.00	1055.63	1008.40	1053.21	2558385.0
2	GOOG	2018-03-23	1047.03	1063.36	1021.22	1021.57	2113497.0	0.0	1.0	1047.03	1063.36	1021.22	1021.57	2113497.0
3	GOOG	2018-03-22	1081.88	1082.90	1045.91	1049.08	2580374.0	0.0	1.0	1081.88	1082.90	1045.91	1049.08	2580374.0
4	GOOG	2018-03-21	1092.74	1106.30	1085.15	1090.88	1640709.0	0.0	1.0	1092.74	1106.30	1085.15	1090.88	1640709.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	GOOG	2014-04-11	532.55	540.00	526.53	530.60	3914100.0	0.0	1.0	532.55	540.00	526.53	530.60	3914100.0
996	GOOG	2014-04-10	565.00	565.00	539.90	540.95	4025800.0	0.0	1.0	565.00	565.00	539.90	540.95	4025800.0
997	GOOG	2014-04-09	559.62	565.37	552.95	564.14	3321700.0	0.0	1.0	559.62	565.37	552.95	564.14	3321700.0
998	GOOG	2014-04-08	542.60	555.00	541.61	554.90	3142600.0	0.0	1.0	542.60	555.00	541.61	554.90	3142600.0
999	GOOG	2014-04-07	540.74	548.48	527.15	538.15	4389600.0	0.0	1.0	540.74	548.48	527.15	538.15	4389600.0

1000 rows × 14 columns

5. Converting date field to DateTime

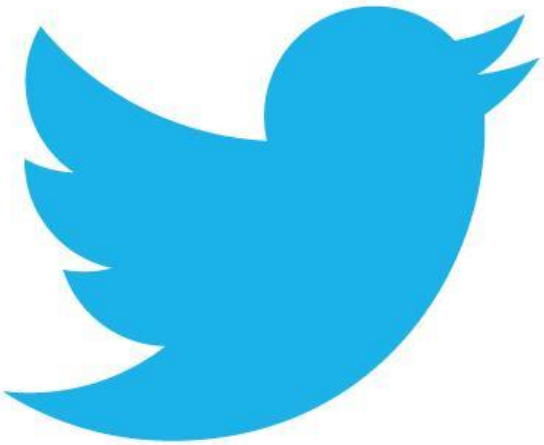
	Company	date	open	high	low	close	volume	adj_close	Year
0	GOOG	2018-03-27	1063.00	1064.84	996.92	1005.10	3029471.0	1005.10	2018
1	GOOG	2018-03-26	1046.00	1055.63	1008.40	1053.21	2558385.0	1053.21	2018
2	GOOG	2018-03-23	1047.03	1063.36	1021.22	1021.57	2113497.0	1021.57	2018
3	GOOG	2018-03-22	1081.88	1082.90	1045.91	1049.08	2580374.0	1049.08	2018
4	GOOG	2018-03-21	1092.74	1106.30	1085.15	1090.88	1640709.0	1090.88	2018

Final Thoughts or conclusion for Quandl API:

After performing these operations I have a good understanding of the service offered by Quandl. The company continues to grow its access to datasets, particularly alternative data,

that can support your trading and investing. The gathered data from Quandl is now ready for **Analysis and Visualisation**.

2. Twitter API:



With **Twitter's API** platform, you'll find endpoints to unlock the data from Tweets, so you can build great experiences and solutions for your customers. These endpoints enable you to manage your Tweets, publish and curate Tweets, filter and search for Tweet topics or trends, and much more.

In order to get data from Twitter you need to register for App.

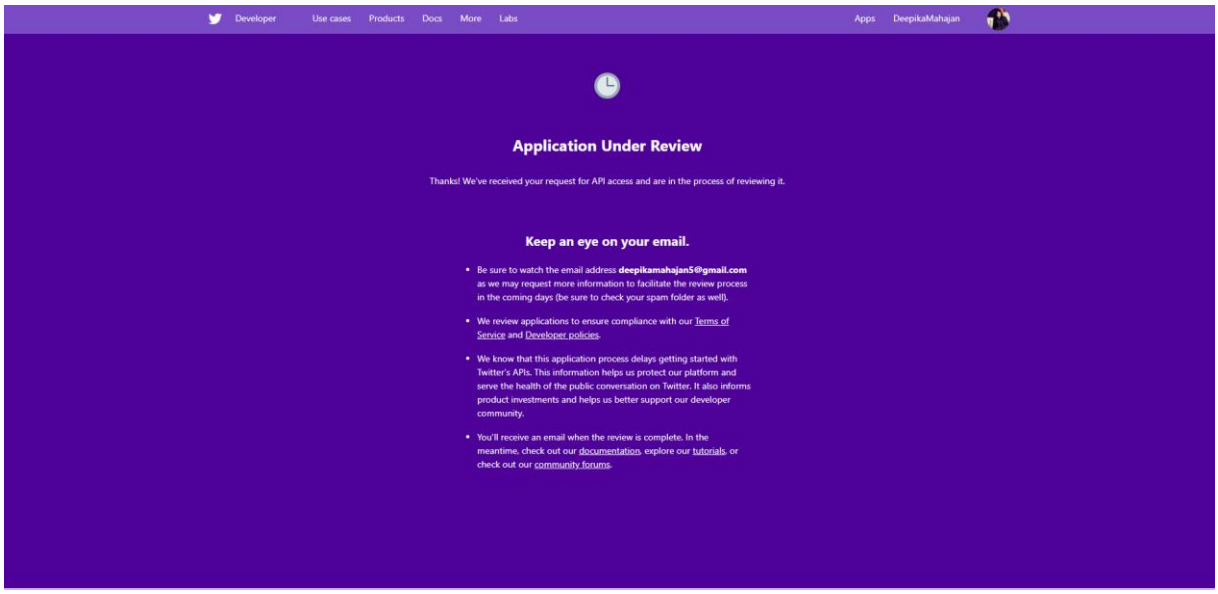
Register Your App

Step1: The first step is the registration of your app.

Step2: In particular, you need to point your browser to http://apps.twitter.com, log-in to Twitter (if you're not already logged in) and register a new application.

Step3: You can now choose a name and a description for your app.

Step4: You will receive a *consumer key* and a *consumer secret*: these are application settings that should always be kept private. For me this step is pending. Twitter is showing me this result. There is no response from them so far.



Step5: After getting the key, from the configuration page of your app, you can also require an access token and an access token secret. Similarly to the consumer keys, these strings must also be kept private: they provide the application access to Twitter on behalf of your account.

Accessing the Data with Twitter API:

Tweepy in one of the most interesting and straightforward to use.

Just because I could not get access to the Twitter App account. I am demonstrating sample code to explain the next steps for data gathering.

```
#Install twitter API
!pip install tweepy==3.3.0
```

```
Requirement already satisfied: tweepy==3.3.0 in /usr/local/lib/python3.6/dist-packages (3.3.0)
Requirement already satisfied: requests>=2.4.3 in /usr/local/lib/python3.6/dist-packages (from tweepy==3.3.0) (2.21.0)
Requirement already satisfied: requests-oauthlib>=0.4.1 in /usr/local/lib/python3.6/dist-packages (from tweepy==3.3.0) (1.3.0)
Requirement already satisfied: six>=1.7.3 in /usr/local/lib/python3.6/dist-packages (from tweepy==3.3.0) (1.12.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.6/dist-packages (from requests>=2.4.3->tweepy==3.3.0) (2019.11.28)
Requirement already satisfied: idna<2.9,>=2.5 in /usr/local/lib/python3.6/dist-packages (from requests>=2.4.3->tweepy==3.3.0) (2.8)
Requirement already satisfied: urllib3<1.25,>=1.21.1 in /usr/local/lib/python3.6/dist-packages (from requests>=2.4.3->tweepy==3.3.0) (1.24.3)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /usr/local/lib/python3.6/dist-packages (from requests>=2.4.3->tweepy==3.3.0) (3.0.4)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.6/dist-packages (from requests-oauthlib>=0.4.1->tweepy==3.3.0) (3.1.0)
```

```
[2] #In order to authorise the app to access Twitter on our behalf, we need to use the OAuth interface:
    #I am writting dummy code just to demonstrate How Twitter API works
    import tweepy
    from tweepy import OAuthHandler

    consumer_key = 'YOUR-CONSUMER-KEY'
    consumer_secret = 'YOUR-CONSUMER-SECRET'
    access_token = 'YOUR-ACCESS-TOKEN'
    access_secret = 'YOUR-ACCESS-SECRET'

    auth = OAuthHandler(consumer_key, consumer_secret)
    auth.set_access_token(access_token, access_secret)

    api = tweepy.API(auth)
```

https://colab.research.google.com/drive/1SAdFio-j6YZ3CMcfKZD5MIZFSoaPOno3

In case of Streaning data we use : StreamListener() to customise the way we process the incoming data. A working example that gathers all the new tweets with the #python hashtag:

```
[9]  #For  Stream Data
     from tweepy import Stream
     from tweepy.streaming import StreamListener

     class MyListener(StreamListener):

         def on_data(self, data):
             try:
                 with open('python.json', 'a') as f:
                     f.write(data)
                     return True
             except BaseException as e:
                 print("Error on_data: %s" % str(e))
             return True

         def on_error(self, status):
             print(status)
             return True

     twitter_stream = Stream(auth, MyListener())
     twitter_stream.filter(track=['#python'])
```

Sources and References:

https://towardsdatascience.com/best-5-free-stock-market-apis-in-2019-ad91dddec984

https://docs.quandl.com/

https://www.geeksforgeeks.org/coding-the-financial-market/

https://decodingmarkets.com/