



Report on Data as a Service (DaaS) & Web Application



JUNE 17, 2017

TEAM 7: SNIGHDHA JOSHI & VIPRA SHAH
UNDER THE GUIDANCE OF – PROF SRI KRISHNAMURTHY

urllib is a python module used for opening HTTP URLs.

- It accomplish tasks such as basic authentication, getting cookies, serving GET/POST requests, error handling, viewing headers.

Get current temperature of a city using Wunderground API

In [2]: `import urllib.request` #import extensible library for opening URL

```
f = urllib.request.urlopen('http://api.wunderground.com/api/a13e2cb8b6838af0/geolookup/conditions/q/MA/Boston.json')
json_string = f.read()
parsed_json = json.loads(json_string)
location = parsed_json['location']['city']
temp_f = parsed_json['current_observation']['temp_f']
print("Current temperature in %s is: %s" % (location, temp_f))
f.close()
```

Current temperature in Boston is: 59.5

Get today's forecast in Fahrenheit of a city using Wunderground API

In [3]: `import urllib.request, json`

```
f = urllib.request.urlopen('http://api.wunderground.com/api/a13e2cb8b6838af0/forecast/q/MA/Boston.json')
json_string = f.read() # Read JSON data
parsed_json = json.loads(json_string)
time = parsed_json['forecast']['txt_forecast']
for each in time['forecastday']:
    print(each['title'] + ': ' + each['fcttext'])
f.close()
```

Friday: Showers possible. Lows overnight in the upper 50s.

Friday Night: Considerable cloudiness with occasional rain showers. Low 58F. Winds SE at 10 to 15 mph. Chance of rain 100%.

Saturday: Rain showers early with overcast skies later in the day. High 77F. Winds ESE at 10 to 15 mph. Chance of rain 50%.

Saturday Night: Partly cloudy skies early will become overcast later during the night. Low 68F. Winds SSW at 10 to 15 mph.

Sunday: Overcast. High 82F. Winds SSW at 15 to 25 mph.

Sunday Night: Partly cloudy skies in the evening, then becoming cloudy overnight. Low around 70F. Winds SSW at 15 to 25 mph.

Monday: Cloudy skies early, then partly cloudy in the afternoon. A stray shower or thunderstorm is possible. High 82F. Winds SS W at 15 to 25 mph.

Monday Night: Scattered thunderstorms developing overnight. Low around 70F. Winds SSW at 15 to 25 mph. Chance of rain 60%.

Quandl

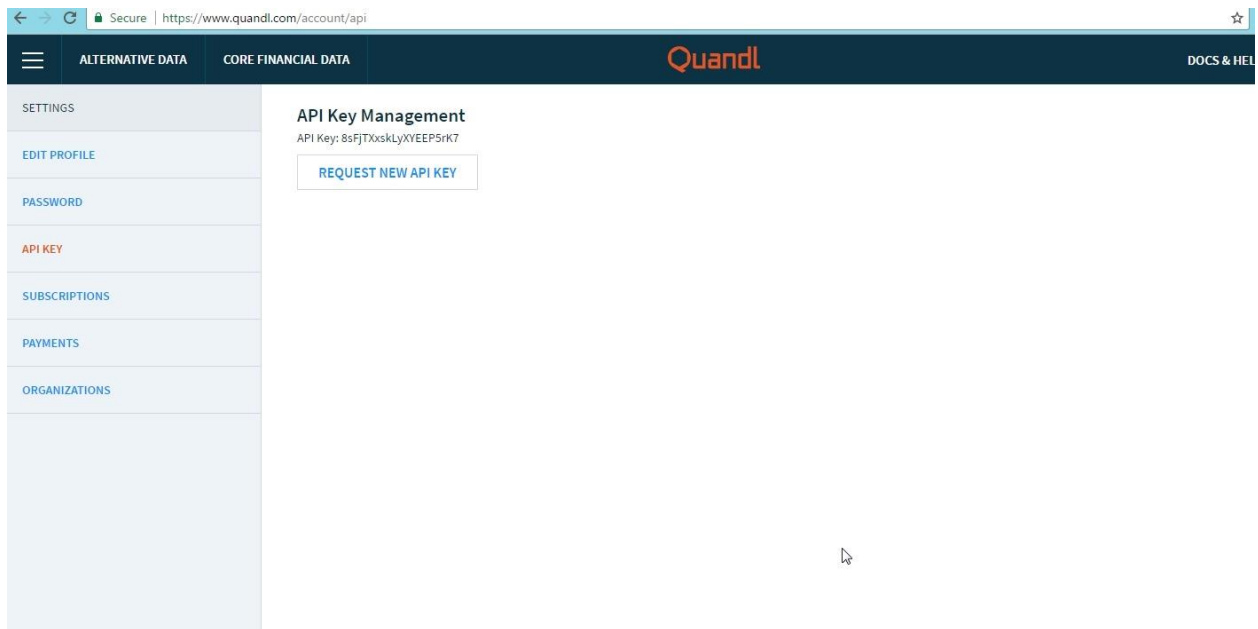
Quandl unifies financial and economic datasets from hundreds of publishers on a single user-friendly platform.

Quandl supports two data formats: time-series and "datatables" (used for non-time-series data).

Import quandl

To get an API key from quandl:

1. Go to <https://www.quandl.com/>
2. Sign up for free
3. Confirmed entered email address, and you should be able to get API keys from API left sidebar.



Matplotlib: is a plotting library for the Python programming language and

Pandas: is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive.

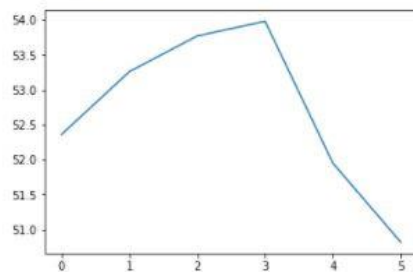
Import pandas as pd

Get information about WTI Crude Oil Price using Quandl dataset

```
In [17]: import quandl
import matplotlib.pyplot as plt
import pandas as pd

quandl.ApiConfig.api_key = "USZdxxsq3v1bxsSZrnsx"
data = quandl.get("EIA/PET_RWTC_D", start_date="2017-01-01", end_date="2017-01-10", returns="numpy") #Filter data using date range
df=pd.DataFrame(data, columns = ['Date', 'Value'])
print(df)
plt.plot(df.Value)
plt.show()
```

	Date	Value
0	2017-01-03	52.36
1	2017-01-04	53.26
2	2017-01-05	53.77
3	2017-01-06	53.98
4	2017-01-09	51.95
5	2017-01-10	50.82



Yahoo_finance

Python module to get stock data from Yahoo! Finance

Get Open and Previous Closing Price of a Stock using yahoo_finance python Package

```
In [18]: from yahoo_finance import Share

yahoo = Share('YHOO') #Use stock code to get data
print(yahoo.get_open())
print(yahoo.get_prev_close())
```

```
52.17
52.63
```

Fred

Python wrapper of the St. Louis Federal Reserve Bank's [FRED API web service] (<http://api.stlouisfed.org/docs/fred/>) for retrieving economic data.

Using information using Python API for FRED (Federal Reserve Economic Data)

```
In [19]: from fred import Fred

fr = Fred(api_key='fb0432249a13111c11d8c2157b3001e3', response_type='dict')
params = {
    'limit': 2,
    'tag_names': 'trade;goods'
}
res = fr.category.series(125, params=params)

for record in res:
    print(record)

{'frequency': 'Monthly', 'frequency_short': 'M', 'id': 'BOPGSTB', 'last_updated': Timestamp('2017-06-02 14:11:03'), 'observation_end': Timestamp('2017-04-01 00:00:00'), 'observation_start': Timestamp('1992-01-01 00:00:00'), 'popularity': 66, 'realtime_end': Timestamp('2017-06-16 00:00:00'), 'realtime_start': Timestamp('2017-06-16 00:00:00'), 'seasonal_adjustment': 'Seasonally Adjusted', 'seasonal_adjustment_short': 'SA', 'title': 'Trade Balance: Goods and Services, Balance of Payments Basis', 'units': 'Millions of Dollars', 'units_short': 'Mil. of $'}
{'frequency': 'Monthly', 'frequency_short': 'M', 'id': 'BOPGTB', 'last_updated': Timestamp('2017-06-02 14:11:12'), 'observation_end': Timestamp('2017-04-01 00:00:00'), 'observation_start': Timestamp('1992-01-01 00:00:00'), 'popularity': 41, 'realtime_end': Timestamp('2017-06-16 00:00:00'), 'realtime_start': Timestamp('2017-06-16 00:00:00'), 'seasonal_adjustment': 'Seasonally Adjusted', 'seasonal_adjustment_short': 'SA', 'title': 'Trade Balance: Goods, Balance of Payments Basis', 'units': 'Millions of Dollars', 'units_short': 'Mil. of $'}
```

Plotly provides a web-service for hosting graphs!

Create a free account to get started. Graphs are saved inside your online Plotly account and you control the privacy.

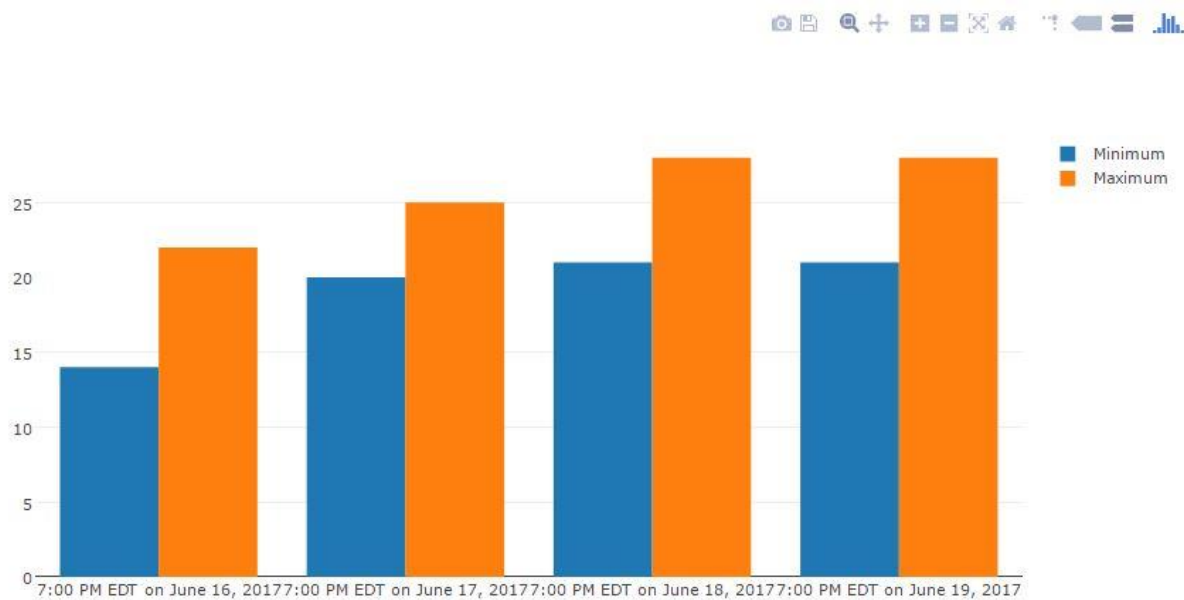
Get Minimum and Maximum temperature in celcius of Next days 3 and plot group bar chart

```
In [14]: import urllib.request, json
import plotly as py
from plotly.graph_objs import *
py.offline.init_notebook_mode() #Run this command at the start of every ipython notebook to use plotly.offline

f = urllib.request.urlopen('http://api.wunderground.com/api/a13e2cb8b6838af0/forecast/q/MA/Boston.json')
json_string = f.read()
parsed_json = json.loads(json_string)
time = parsed_json['forecast']['simpleforecast']['forecastday']
low = []
high = []
day = []
for each in time:
    low.append(each['low']['celsius'])
    high.append(each['high']['celsius'])
    day.append(each['date']['pretty'])

# Plot group bar chart for Minimum and Maximum temperature against Day
group1 = go.Bar(x=day, y=low, name='Minimum')
group2 = go.Bar(x=day, y=high, name='Maximum')
data = [group1, group2]
layout = go.Layout(barmode='group')

fig = go.Figure(data=data, layout=layout)
py.offline.iplot(fig, filename='jupyter/group-bar-chart')
f.close()
```



Flask and Python

For the deployment of web application, we will need:

- 1) Virtual environment
- 2) Flask framework

Following is the step by step guide to install both:

- 1) Virtual environment

`pip install virtualenv`

`sudo pip install virtualenv` – **For Mac OS and Linux Users**

Once you have virtualenv installed, create your own environment. create a project folder and a venv folder within:

`mkdir myproject`

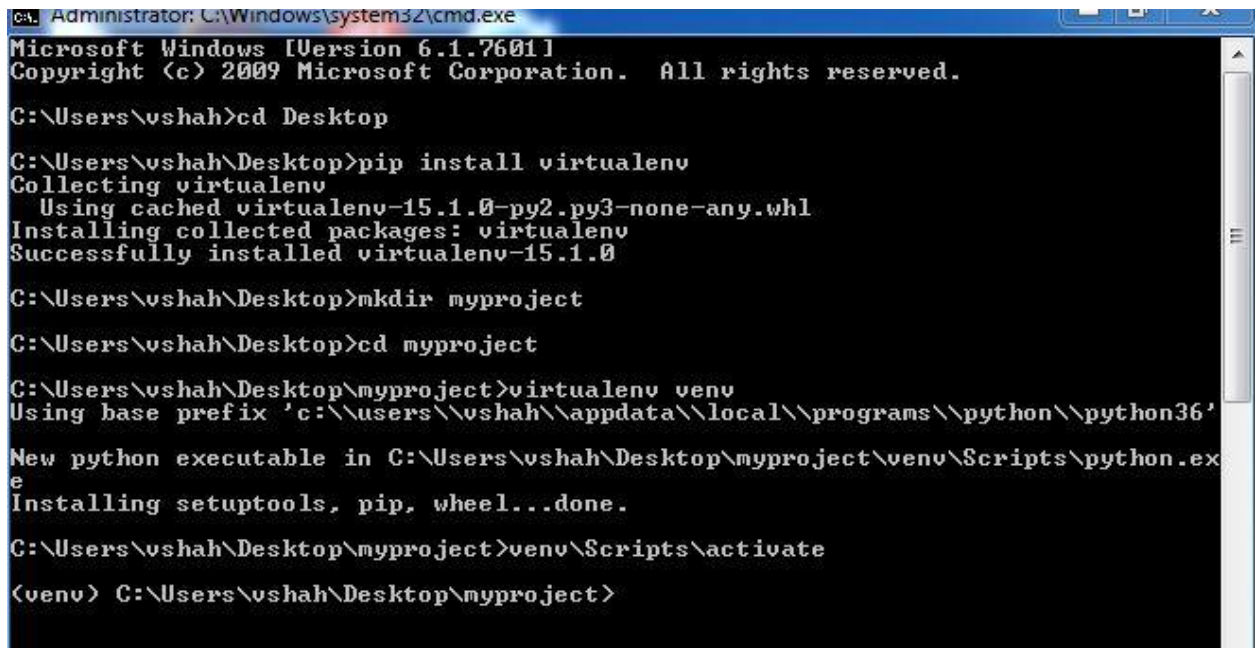
`cd myproject`

`virtualenv venv`

`venv\Scripts\activate`

`$. venv/bin/activate` - **For Mac OS and Linux users**

To confirm that you are into virtual environment, you should be able to see "<venv>" before path.



```
ca. Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\vshah>cd Desktop

C:\Users\vshah\Desktop>pip install virtualenv
Collecting virtualenv
  Using cached virtualenv-15.1.0-py2.py3-none-any.whl
Installing collected packages: virtualenv
Successfully installed virtualenv-15.1.0

C:\Users\vshah\Desktop>mkdir myproject
C:\Users\vshah\Desktop>cd myproject

C:\Users\vshah\Desktop\myproject>virtualenv venv
Using base prefix 'c:\users\vshah\appdata\local\programs\python\python36'
New python executable in C:\Users\vshah\Desktop\myproject\venv\Scripts\python.exe
Installing setuptools, pip, wheel...done.

C:\Users\vshah\Desktop\myproject>venv\Scripts\activate

(venv) C:\Users\vshah\Desktop\myproject>
```

2) Flask Framework

Flask is a micro web framework written in Python and based on the Werkzeug toolkit and Jinja2 template engine.

Pip install Flask


```
(venv) C:\Users\vshah\Desktop\myproject>pip install Flask
Collecting Flask
  Using cached Flask-0.12.2-py2.py3-none-any.whl
Collecting click>=2.0 (from Flask)
  Using cached click-6.7-py2.py3-none-any.whl
Collecting Werkzeug>=0.7 (from Flask)
  Using cached Werkzeug-0.12.2-py2.py3-none-any.whl
Collecting itsdangerous>=0.21 (from Flask)
  Using cached itsdangerous-0.24.tar.gz
Collecting Jinja2>=2.4 (from Flask)
  Using cached Jinja2-2.9.6-py2.py3-none-any.whl
Collecting MarkupSafe>=0.23 (from Jinja2>=2.4->Flask)
  Using cached MarkupSafe-1.0.tar.gz
Building wheels for collected packages: itsdangerous, MarkupSafe
  Running setup.py bdist_wheel for itsdangerous ... done
  Stored in directory: C:\Users\vshah\AppData\Local\pip\Cache\wheels\fc\8a\66\24\d655233c757e178d45dea2de22a04c6d92766abfb741129a
  Running setup.py bdist_wheel for MarkupSafe ... done
  Stored in directory: C:\Users\vshah\AppData\Local\pip\Cache\wheels\88\8a\7\30\ee39a54a87bcb25308fa3ca64e8ddc75d9b3e5afa21ee32d57
Successfully built itsdangerous MarkupSafe
Installing collected packages: click, Werkzeug, itsdangerous, MarkupSafe, Jinja2, Flask
Successfully installed Flask-0.12.2 Jinja2-2.9.6 MarkupSafe-1.0 Werkzeug-0.12.2 click-6.7 itsdangerous-0.24

(venv) C:\Users\vshah\Desktop\myproject>
```

To Run web application only follow this steps:

- 1) Get the resources and code from Github repository.
- 2) Either you can copy the files into newly created folder(myproject) or follow the steps to create virtual environment for downloaded folder (web application)
- 3) Now, run pip install -r requirements.txt

This will install all the dependency modules used in application such as Flask, yahoo_finance. (So, you can skip the step to install Flask, this command will install all the packages used in the application)

```

(venv) C:\Users\vshah\Desktop\myproject>pip install -r requirements.txt
Requirement already satisfied: Flask==0.12.2 in c:\users\vshah\desktop\myproject\venv\lib\site-packages (from -r requirements.txt (line 1))
Collecting yahoo-finance==1.4.0 (from -r requirements.txt (line 2))
  Using cached yahoo-finance-1.4.0.tar.gz
Requirement already satisfied: Werkzeug==0.7 in c:\users\vshah\desktop\myproject\venv\lib\site-packages (from Flask==0.12.2->-r requirements.txt (line 1))
Requirement already satisfied: Jinja2==2.4 in c:\users\vshah\desktop\myproject\venv\lib\site-packages (from Flask==0.12.2->-r requirements.txt (line 1))
Requirement already satisfied: click==2.0 in c:\users\vshah\desktop\myproject\venv\lib\site-packages (from Flask==0.12.2->-r requirements.txt (line 1))
Requirement already satisfied: itsdangerous==0.21 in c:\users\vshah\desktop\myproject\venv\lib\site-packages (from Flask==0.12.2->-r requirements.txt (line 1))
Collecting pytz (from yahoo-finance==1.4.0->-r requirements.txt (line 2))
  Using cached pytz-2017.2-py2.py3-none-any.whl
Collecting simplejson (from yahoo-finance==1.4.0->-r requirements.txt (line 2))
  Using cached simplejson-3.10.0.tar.gz
Requirement already satisfied: MarkupSafe==0.23 in c:\users\vshah\desktop\myproject\venv\lib\site-packages (from Jinja2==2.4->Flask==0.12.2->-r requirements.txt (line 1))
Building wheels for collected packages: yahoo-finance, simplejson
  Running setup.py bdist_wheel for yahoo-finance ... done
  Stored in directory: C:\Users\vshah\AppData\Local\pip\Cache\wheels\8b\32\7a\1f\3e589e823ef2b8e510a99d11e3abae8f28af2ab8d4eaa884
  Running setup.py bdist_wheel for simplejson ... done
  Stored in directory: C:\Users\vshah\AppData\Local\pip\Cache\wheels\43\c5\ef\ec\cebbb19becffd2ba75bf219afdbb4ca85198b2d909f1b31b
Successfully built yahoo-finance simplejson
Installing collected packages: pytz, simplejson, yahoo-finance
Successfully installed pytz-2017.2 simplejson-3.10.0 yahoo-finance-1.4.0

```

4) Now run following command python app.py

```

(venv) C:\Users\vshah\Desktop\myproject>python app.py
* Restarting with stat
* Debugger is active!
* Debugger PIN: 311-118-411
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

```

This will show that application has started on local host, port 5000. Flask uses 5000 as a default port.

Open <http://127.0.0.1:5000/> -- to access web application and you should be able to see this screen. ☺

127.0.0.1:5000/send



Flask is Fun!

Get Stock Data from YAHOO Finance API

Stock Name:

Get Stock Data

Previous closing price for above stock is: 950.76

Year high for above stock is: 988.25

Year Low for above stock is: 663.28

@Copyrights: Vipra and Snigdha