Fitbit Data Capture Using Fitbit API Overview

***Important Links***

<https://towardsdatascience.com/collect-your-own-fitbit-data-with-python-ff145fa10873>\*\* Online blog - some errors in the code posted on this site\*\*

<https://python-fitbit.readthedocs.io/en/latest/> \*\*Fitbit API Package Documentation\*\*

<https://github.com/orcasgit/python-fitbit> \*\*GitHub repository with Fitbit API Python Client\*\*

*Project Setup:*

The first step is to follow the instructions posted in the first link above and to register an app on the Fitbit site. Following this, it is necessary to download the Fitbit API Python Client from Github in the third link above. Then proceed with the directions in the first link.

Using a popular data science platform like Anaconda will likely make the setup for this project easier. The preinstalled packages in Anaconda limit the extra packages needed, and the only non-anaconda package to be installed is CherryPy which can be easily installed using the command *conda install -c anaconda cherrypy*.

***Code with Comments/Explanations***

#Import the necessary packages

import fitbit

import gather\_keys\_oauth2 as Oauth2

import pandas as pd

import datetime

#Both the Client ID and Client Secret come from when Fitbit site after registering an app

CLIENT\_ID = '22BDG8'

CLIENT\_SECRET = '7e5757bedbfcf9cb8cb3ffa11b75331b'

#This gives us access to our own data. It seems like Fitbit does not usually provide access to your own data unless following authorization steps

server = Oauth2.OAuth2Server(CLIENT\_ID, CLIENT\_SECRET)

server.browser\_authorize()

ACCESS\_TOKEN = str(server.fitbit.client.session.token['access\_token'])

REFRESH\_TOKEN = str(server.fitbit.client.session.token['refresh\_token'])

auth2\_client = fitbit.Fitbit(CLIENT\_ID, CLIENT\_SECRET, oauth2=True, access\_token=ACCESS\_TOKEN,

refresh\_token=REFRESH\_TOKEN)

#This defines the dates that we want to pull our data from using the datetime package in python

yesterday = str((datetime.datetime.now() - datetime.timedelta(days=1)).strftime("%Y%m%d"))

yesterday2 = str((datetime.datetime.now() - datetime.timedelta(days=1)).strftime("%Y-%m-%d"))

today = str(datetime.datetime.now().strftime("%Y%m%d"))

#This line retrieves heart rate data.

fit\_statsHR = auth2\_client.intraday\_time\_series('activities/heart', base\_date=yesterday2, detail\_level='1sec')

#This pulls the data, that is formatted in a list, and places it in a pandas data frame. This may likely need some refining\*\*

# It is possible to export this data as a csv file, which will likely be useful as a future idea/improvement

time\_list = []

val\_list = []

for i in fit\_statsHR['activities-heart']:

val\_list.append(i['value'])

time\_list.append(i['dateTime'])

heartdf = pd.DataFrame({'Heart Rate':val\_list,'Time':time\_list})

#pulls sleep data

fit\_statsSl = auth2\_client.sleep(date='today')

#Same process as above of putting the data into a data frame

stime\_list = []

sval\_list = []

for i in fit\_statsSl['sleep']:

stime\_list.append(i['dateTime'])

sval\_list.append(i['value'])

sleepdf = pd.DataFrame({'State':sval\_list,

'Time':stime\_list})

# activities for all time.

# The steps and floors and distance all match the data on our fitbit account !!!

activities = auth2\_client.activity\_stats(user\_id=None, qualifier='')

activities

***Future Ideas***

* Investigating the rest of the Fitbit API Python Client Documentation to pull more data
* Editing this code so it is more efficient?
* Be able to pull more data than one day at a time
* Export Data in CSV files if needed