

# 데이터과학 실습 보고서

-4주차 fitbit API를 이용한 걸음 수 측정-

2017.3.28

201201185 장진우

# 1. Fitbit O-Auth 2.0을 이용한 인증방법

Step 1.

- sudo apt-get install git 을 통해 git을 다운받는다.
- git clone <https://github.com/orcasgit/python-fitbit>을 하여 API를 다운 받는다.

Step 2.

- pip install cherrypy를 다운 받는다.
- pip install requests를 다운 받는다.
- 그 후 일반 사용자로 실행하면 제대로 실행이 되지 않으므로 root 권한으로 실행한다.

```
jlnwoo@jlnwoo:~$ ls
examples.desktop  공개 문서 비디오 음악
python-fitbit     다운로드 바탕화면 사진 템플릿
jlnwoo@jlnwoo:~$ cd python-fitbit/
jlnwoo@jlnwoo:~/python-fitbit$ ls
AUTHORS          datadumpHeart.json  graph.py          step2.json
CHANGELOG.rst    datadumpSteps.json  graph2.py         steptest.py
LICENSE          docs                heart.py          steptest2.py
MANIFEST.in      fitbit              requirements       tox.ini
README.rst       fitbit_tests        setup.py
config.ini       gather_keys_oauth2.py  step.json
jlnwoo@jlnwoo:~/python-fitbit$ python gather_keys_oauth2.py
Traceback (most recent call last):
  File "gather_keys_oauth2.py", line 10, in <module>
    from fitbit.api import Fitbit
  File "/home/jlnwoo/python-fitbit/fitbit/__init__.py", line 10, in <module>
    from .api import Fitbit, FitbitOAuth2Client
  File "/home/jlnwoo/python-fitbit/fitbit/api.py", line 13, in <module>
    from requests_oauthlib import OAuth2Session
ImportError: cannot import name OAuth2Session
jlnwoo@jlnwoo:~/python-fitbit$
```

<사용자권한으로는 오류가 뜬>

Step 3.

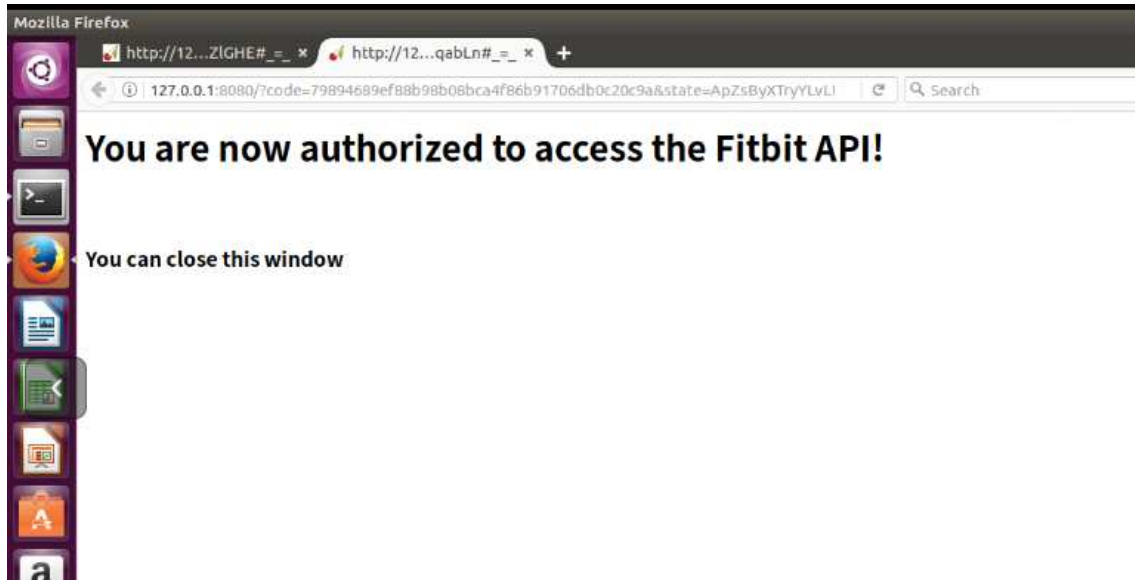
-root 권한으로 변경 후 python gather\_keys\_oauth2.py 파일을 실행시킨다.

```
root@jinwoo: /home/jinwoo/python-fitbit
examples.desktop  공개 문서 비디오 음악
python-fitbit     다운로드 바탕화면 사진 템플릿
jinwoo@jinwoo:~$ cd python-fitbit/
jinwoo@jinwoo:~/python-fitbit$ ls
AUTHORS          datadumpHeart.json      graph.py          step2.json
CHANGELOG.rst    datadumpSteps.json     graph2.py        steptest.py
LICENSE          docs                   heart.py         steptest2.py
MANIFEST.in      fitbit                 requirements     tox.ini
README.rst       fitbit_tests          setup.py
config.ini       gather_keys_oauth2.py  step.json
jinwoo@jinwoo:~/python-fitbit$ python gather_keys_oauth2.py
Traceback (most recent call last):
  File "gather_keys_oauth2.py", line 10, in <module>
    from fitbit.api import Fitbit
  File "/home/jinwoo/python-fitbit/fitbit/__init__.py", line 10, in <module>
    from .api import Fitbit, FitbitOAuth2Client
  File "/home/jinwoo/python-fitbit/fitbit/api.py", line 13, in <module>
    from requests_oauthlib import OAuth2Session
ImportError: cannot import name OAuth2Session
jinwoo@jinwoo:~/python-fitbit$ su root
암호:
root@jinwoo:/home/jinwoo/python-fitbit# python gather_keys_oauth2.py
Arguments: client_id and client_secret
root@jinwoo:/home/jinwoo/python-fitbit#
```

-Arguments :client\_id and client\_secret이라는 경고문이 나올 때 위의 문장 다음으로 client\_id client\_secret을 친다.

```
root@jinwoo: /home/jinwoo/python-fitbit
python-fitbit     다운로드 바탕화면 사진 템플릿
jinwoo@jinwoo:~$ cd python-fitbit/
jinwoo@jinwoo:~/python-fitbit$ ls
AUTHORS          datadumpHeart.json      graph.py          step2.json
CHANGELOG.rst    datadumpSteps.json     graph2.py        steptest.py
LICENSE          docs                   heart.py         steptest2.py
MANIFEST.in      fitbit                 requirements     tox.ini
README.rst       fitbit_tests          setup.py
config.ini       gather_keys_oauth2.py  step.json
jinwoo@jinwoo:~/python-fitbit$ python gather_keys_oauth2.py
Traceback (most recent call last):
  File "gather_keys_oauth2.py", line 10, in <module>
    from fitbit.api import Fitbit
  File "/home/jinwoo/python-fitbit/fitbit/__init__.py", line 10, in <module>
    from .api import Fitbit, FitbitOAuth2Client
  File "/home/jinwoo/python-fitbit/fitbit/api.py", line 13, in <module>
    from requests_oauthlib import OAuth2Session
ImportError: cannot import name OAuth2Session
jinwoo@jinwoo:~/python-fitbit$ su root
암호:
root@jinwoo:/home/jinwoo/python-fitbit# python gather_keys_oauth2.py
Arguments: client_id and client_secret
root@jinwoo:/home/jinwoo/python-fitbit# python gather_keys_oauth2.py 228D5W f340
becbc7dba5665706e1b24189e7fd
```

-모든 부분을 허용하면 Fitbit API를 사용할 준비가 되었다.



Step 4.

-위의 창을 끄게 되면 아래와 같이 refresh\_token과 access\_token이 나오는데 이 token들을 복사한다.

```
You are authorized to access data for the user: 장진우
TOKEN
=====

token_type = Bearer
user_id = 5JX6VB
refresh_token = f3c9b136aa8595e05b518fe0dd5f14d113915a57f62e98162bb84bbd510ad580
access_token = eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1Slg2VkIiLCJhdWQiOiIyMjhENVciLCJp
c3MiOiJGaXRiaXQiLCJ0eXAiOiJhY2Nlc3NfdG9rZW4iLCJzY29wZXMiOiJ3aHIgd251dCB3cHJvIHdz
bGUgd3dlasB3c29jIHdhY3Qgd3NldCB3bG9jIiwiaXhwIjoxNDkwnjYxNjIzLCJpYXQiOiJl00TA2MzI4
MjN9.GflngV8MmcMNZ_RzKybkSufIymvQ30JVDciW9Btfqyw
scope = ['u'activity', 'u'weight', 'u'settings', 'u'nutrition', 'u'profile', 'u'social
', 'u'location', 'u'sleep', 'u'heartrate']
expires_in = 28800
expires_at = 1490661624.15
```

-그 후, config.ini라는 파일을 만들어 아래와 같이 입력한다.

```
[ACCOUNT]
CLIENT_ID =228D5W
CLIENT_SECRET = f340becbc7dba5665706e1b24189e7fd
ACCESS_TOKEN = eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1Slg2VkIiLCJhdWQiOiIyMjhENVciLCJp
c3MiOiJGaXRiaXQiLCJ0eXAiOiJhY2Nlc3NfdG9rZW4iLCJzY29wZXMiOiJ3aHIgd251dCB3cHJvIHdz
bGUgd3dlasB3c29jIHdhY3Qgd2FjdCB3bG9jIiwiaXhwIjoxNDkwnjM1MDIyLCJpYXQiOiJl00TA2MDYy
MjN9.ImCh7LyfTl_YchJDWFLbaXzrgXduuNLSdVXgn1JufI
REFRESH_TOKEN =f3f111b900bc6951826ddf94c630723fa3cf681a8721934990cc0931d1711a8f
~
```



Step 5.

- 예제 코드를 작성하여 Fitbit API를 제대로 사용가능 한지 테스트 해본다.

```
import fitbit
import json
import ConfigParser

config = ConfigParser.RawConfigParser()
config.read('config.ini')

CLIENT_ID = config.get('ACCOUNT','CLIENT_ID')
CLIENT_SECRET = config.get('ACCOUNT','CLIENT_SECRET')
ACCESS_TOKEN = config.get('ACCOUNT','ACCESS_TOKEN')
REFRESH_TOKEN = config.get('ACCOUNT','REFRESH_TOKEN')

authd_client = fitbit.Fitbit(CLIENT_ID, CLIENT_SECRET, access_token=ACCESS_TOKEN, refresh_token= REFRESH_TOKEN)

intraday_step = authd_client.intraday_time_series('activities/steps', base_date='2017-03-26',detail_level = '15min')

print(intraday_step)

f=open('step.json','w')
json.dump(intraday_step,f,encoding="UTF-8")
-
```

- JSON형식의 파일이 잘 작성된 것을 확인한다.

```
{
  "activities-steps-intraday": {
    "datasetType": "minute",
    "datasetInterval": 15,
    "dataset": [
      {
        "value": 0,
        "time": "00:00:00"
      },
      {
        "value": 11,
        "time": "00:15:00"
      },
      {
        "value": 0,
        "time": "00:30:00"
      },
      {
        "value": 0,
        "time": "00:45:00"
      },
      {
        "value": 0,
        "time": "01:00:00"
      },
      {
        "value": 0,
        "time": "01:15:00"
      },
      {
        "value": 4,
        "time": "01:30:00"
      },
      {
        "value": 0,
        "time": "01:45:00"
      },
      {
        "value": 321,
        "time": "02:00:00"
      },
      {
        "value": 0,
        "time": "02:15:00"
      },
      {
        "value": 4,
        "time": "02:30:00"
      },
      {
        "value": 20,
        "time": "02:45:00"
      },
      {
        "value": 0,
        "time": "03:00:00"
      },
      {
        "value": 0,
        "time": "03:15:00"
      },
      {
        "value": 0,
        "time": "03:30:00"
      },
      {
        "value": 29,
        "time": "03:45:00"
      },
      {
        "value": 22,
        "time": "04:00:00"
      },
      {
        "value": 0,
        "time": "04:15:00"
      },
      {
        "value": 0,
        "time": "04:30:00"
      },
      {
        "value": 0,
        "time": "04:45:00"
      },
      {
        "value": 0,
        "time": "05:00:00"
      },
      {
        "value": 0,
        "time": "05:15:00"
      },
      {
        "value": 0,
        "time": "05:30:00"
      },
      {
        "value": 0,
        "time": "05:45:00"
      },
      {
        "value": 0,
        "time": "06:00:00"
      },
      {
        "value": 0,
        "time": "06:15:00"
      },
      {
        "value": 0,
        "time": "06:30:00"
      },
      {
        "value": 0,
        "time": "06:45:00"
      },
      {
        "value": 0,
        "time": "07:00:00"
      },
      {
        "value": 0,
        "time": "07:15:00"
      },
      {
        "value": 0,
        "time": "07:30:00"
      },
      {
        "value": 0,
        "time": "07:45:00"
      },
      {
        "value": 0,
        "time": "08:00:00"
      },
      {
        "value": 0,
        "time": "08:15:00"
      },
      {
        "value": 0,
        "time": "08:30:00"
      },
      {
        "value": 0,
        "time": "08:45:00"
      },
      {
        "value": 7,
        "time": "09:00:00"
      },
      {
        "value": 0,
        "time": "09:15:00"
      },
      {
        "value": 0,
        "time": "09:30:00"
      },
      {
        "value": 0,
        "time": "09:45:00"
      },
      {
        "value": 0,
        "time": "10:00:00"
      },
      {
        "value": 0,
        "time": "10:15:00"
      },
      {
        "value": 0,
        "time": "10:30:00"
      },
      {
        "value": 0,
        "time": "10:45:00"
      },
      {
        "value": 0,
        "time": "11:00:00"
      },
      {
        "value": 0,
        "time": "11:15:00"
      },
      {
        "value": 0,
        "time": "11:30:00"
      },
      {
        "value": 0,
        "time": "11:45:00"
      },
      {
        "value": 0,
        "time": "12:00:00"
      },
      {
        "value": 24,
        "time": "12:15:00"
      },
      {
        "value": 0,
        "time": "12:30:00"
      },
      {
        "value": 73,
        "time": "12:45:00"
      },
      {
        "value": 42,
        "time": "13:00:00"
      },
      {
        "value": 16,
        "time": "13:15:00"
      },
      {
        "value": 0,
        "time": "13:30:00"
      },
      {
        "value": 33,
        "time": "13:45:00"
      },
      {
        "value": 277,
        "time": "14:00:00"
      },
      {
        "value": 9,
        "time": "14:15:00"
      },
      {
        "value": 0,
        "time": "14:30:00"
      },
      {
        "value": 0,
        "time": "14:45:00"
      },
      {
        "value": 4,
        "time": "15:00:00"
      },
      {
        "value": 0,
        "time": "15:15:00"
      },
      {
        "value": 6,
        "time": "15:30:00"
      },
      {
        "value": 67,
        "time": "15:45:00"
      },
      {
        "value": 9,
        "time": "16:00:00"
      },
      {
        "value": 80,
        "time": "16:15:00"
      },
      {
        "value": 83,
        "time": "16:30:00"
      },
      {
        "value": 23,
        "time": "16:45:00"
      },
      {
        "value": 0,
        "time": "17:00:00"
      },
      {
        "value": 0,
        "time": "17:15:00"
      },
      {
        "value": 62,
        "time": "17:30:00"
      },
      {
        "value": 45,
        "time": "17:45:00"
      },
      {
        "value": 0,
        "time": "18:00:00"
      },
      {
        "value": 0,
        "time": "18:15:00"
      },
      {
        "value": 31,
        "time": "18:30:00"
      },
      {
        "value": 190,
        "time": "18:45:00"
      },
      {
        "value": 298,
        "time": "19:00:00"
      },
      {
        "value": 24,
        "time": "19:15:00"
      },
      {
        "value": 470,
        "time": "19:30:00"
      },
      {
        "value": 265,
        "time": "19:45:00"
      },
      {
        "value": 7,
        "time": "20:00:00"
      },
      {
        "value": 0,
        "time": "20:15:00"
      },
      {
        "value": 0,
        "time": "20:30:00"
      },
      {
        "value": 0,
        "time": "20:45:00"
      },
      {
        "value": 0,
        "time": "21:00:00"
      },
      {
        "value": 0,
        "time": "21:15:00"
      },
      {
        "value": 594,
        "time": "21:30:00"
      },
      {
        "value": 1197,
        "time": "21:45:00"
      },
      {
        "value": 486,
        "time": "22:00:00"
      },
      {
        "value": 430,
        "time": "22:15:00"
      },
      {
        "value": 57,
        "time": "22:30:00"
      },
      {
        "value": 36,
        "time": "22:45:00"
      },
      {
        "value": 28,
        "time": "23:00:00"
      },
      {
        "value": 6,
        "time": "23:15:00"
      },
      {
        "value": 33,
        "time": "23:30:00"
      },
      {
        "value": 0,
        "time": "23:45:00"
      }
    ]
  },
  "activities-steps": [
    {
      "value": 5423,
      "dateTime": "2017-03-26"
    }
  ]
}
```

## 2. 다운로드한 데이터로 그래프 그리기

### 1.intraday

```
import json
import datetime
import matplotlib.pyplot as plt
from datetime import datetime

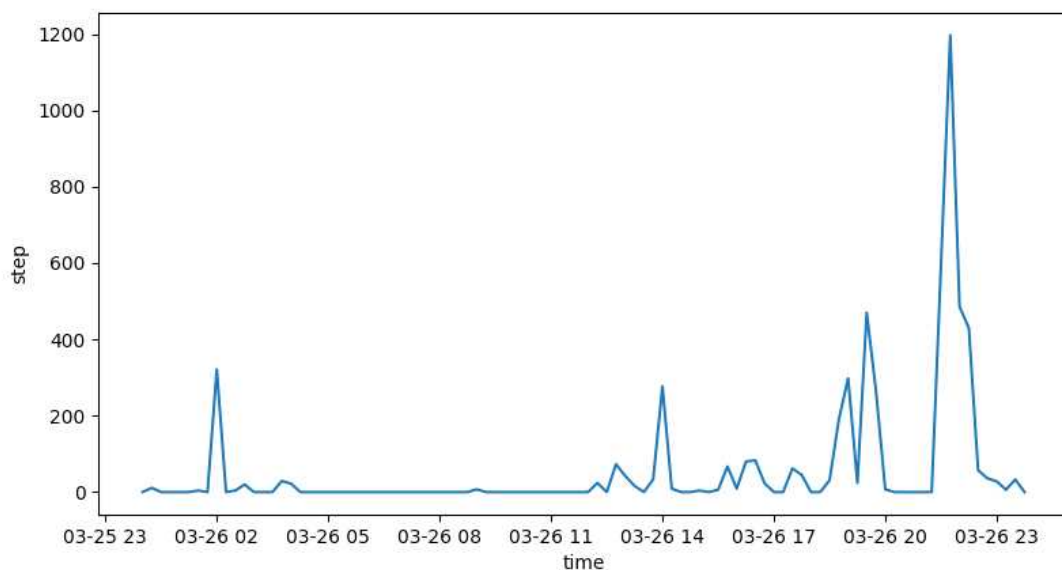
json_file = "./step.json"
result={}
def readConfig(filename):
    f = open(filename,'r')
    js = json.loads(f.read())
    f.close()
    return js
def main():
    global json_file
    global result

    result = readConfig(json_file)
    values=[]
    times=[]
    length=len(result['activities-steps-intraday']['dataset'])
    for i in range(length):
        values.append([result['activities-steps-intraday']['dataset'][i]['value']])

        temp=result['activities-steps-intraday']['dataset'][i]['time']
        change=datetime(2017,03,26,int(temp[0]+temp[1]),int(temp[3]+temp[4]),int(temp[6]+temp[7]))
        times.append([change])

    print(times)
    plt.plot(times,values)
    plt.xlabel("time")
    plt.ylabel("step")
    plt.show()
if __name__=="__main__":
    main()
```

우선 json으로 만든 intraday data를 파싱하여 result라는 변수에 넣습니다. 그 후 result의['dataset']의 원소만큼 for문을 반복하여 value값과 time값을 분리시킵니다. 그 후, plot함수를 통해 그래프를 그리게 됩니다.



<intraday data 'step for 15min'을 이용한 그래프>

## 2. 2주간의 걸음수

```
import fitbit
import json
import ConfigParser

config = ConfigParser.RawConfigParser()
config.read('config.ini')

CLIENT_ID = config.get('ACCOUNT', 'CLIENT_ID')
CLIENT_SECRET = config.get('ACCOUNT', 'CLIENT_SECRET')
ACCESS_TOKEN = config.get('ACCOUNT', 'ACCESS_TOKEN')
REFRESH_TOKEN = config.get('ACCOUNT', 'REFRESH_TOKEN')

authd_client = fitbit.Fitbit(CLIENT_ID, CLIENT_SECRET, access_token=ACCESS_TOKEN, refresh_token=REFRESH_TOKEN)

week_step=[]
days=['2017-03-13','2017-03-14','2017-03-15','2017-03-16','2017-03-17','2017-03-18','2017-03-19','2017-03-20','2017-03-21','2017-03-22',
'2017-03-23','2017-03-24','2017-03-25','2017-03-26']
for i in range(14):
    intraday_step = authd_client.intraday_time_series('activities-steps', base_date=days[i])
    week_step.append(intraday_step['activities-steps'])

print(week_step)

f=open('step2.json','w')
json.dump(week_step,f,encoding="UTF-8")
```

앞서 구하였던 intraday의 총합인 activities-steps의 원소를 따로 추출하여 저장한 후 json 파일로 만듭니다.

```
import json
import datetime
import matplotlib.pyplot as plt
from datetime import datetime

json_file = "./step2.json"
result={}
def readConfig(filename):
    f = open(filename, 'r')
    js = json.loads(f.read())
    f.close()
    return js

def main():
    global json_file
    global result

    result = readConfig(json_file)

    values=[]
    days=[]

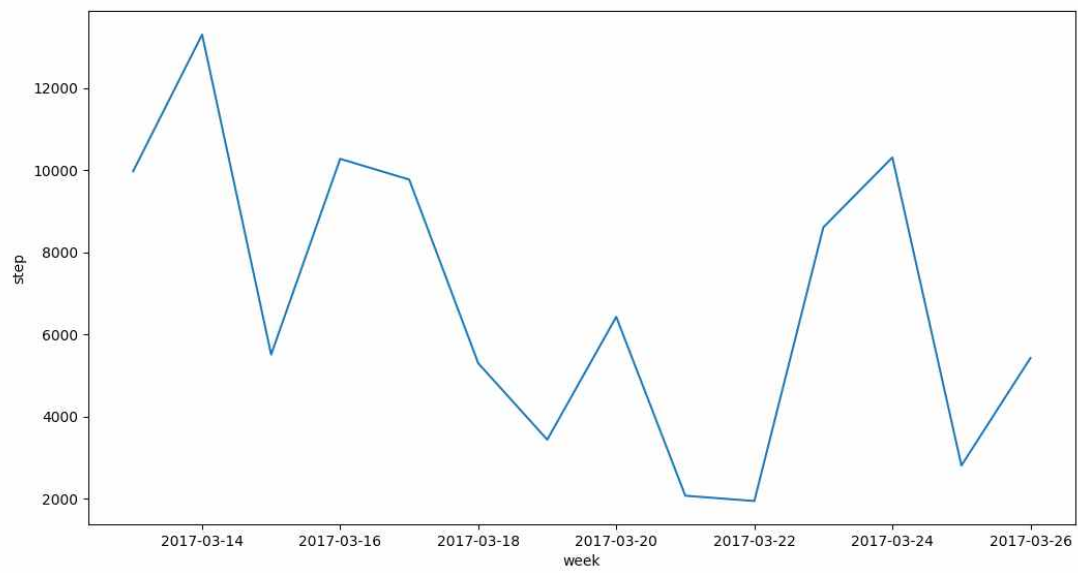
    length=len(result)
    for i in range(length):
        values.append(result[i][0]['value'].encode("utf-8"))
        days.append(datetime.strptime(result[i][0]['dateTime'], "%Y-%m-%d"))

    print(values)
    plt.plot(days, values)
    plt.xlabel("week")
    plt.ylabel("step")

    plt.show()

if __name__ == "__main__":
```

앞에서 추출한 데이터를 파싱하여 value값과 day값을 추출한 뒤 plot함수를 이용하여 그래프를 그리게 됩니다.



<2주간의 걸음 수 그래프>