데이터과학

...

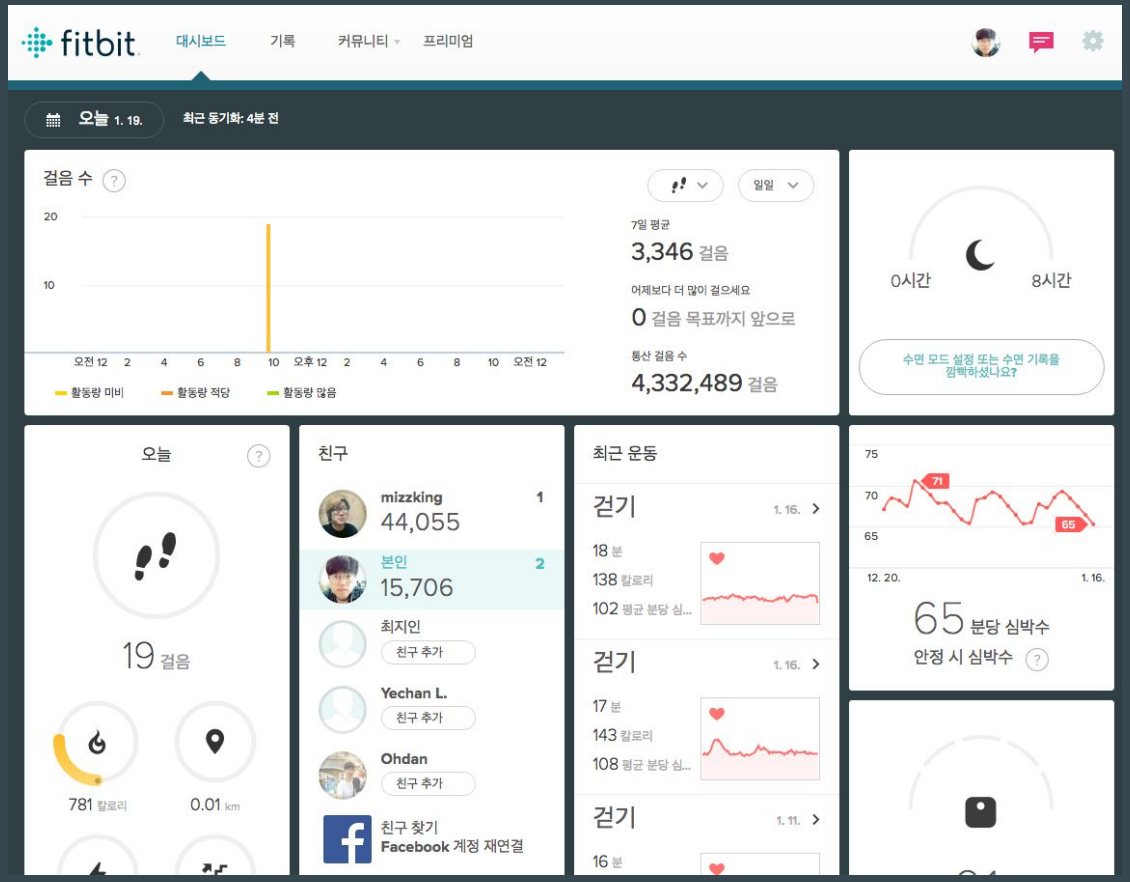
4주차

지난주

- 제출률 : 32/48
  - 66.7%
- 질문
  - 11개
- 저번주와 비슷
- 채점 아직..

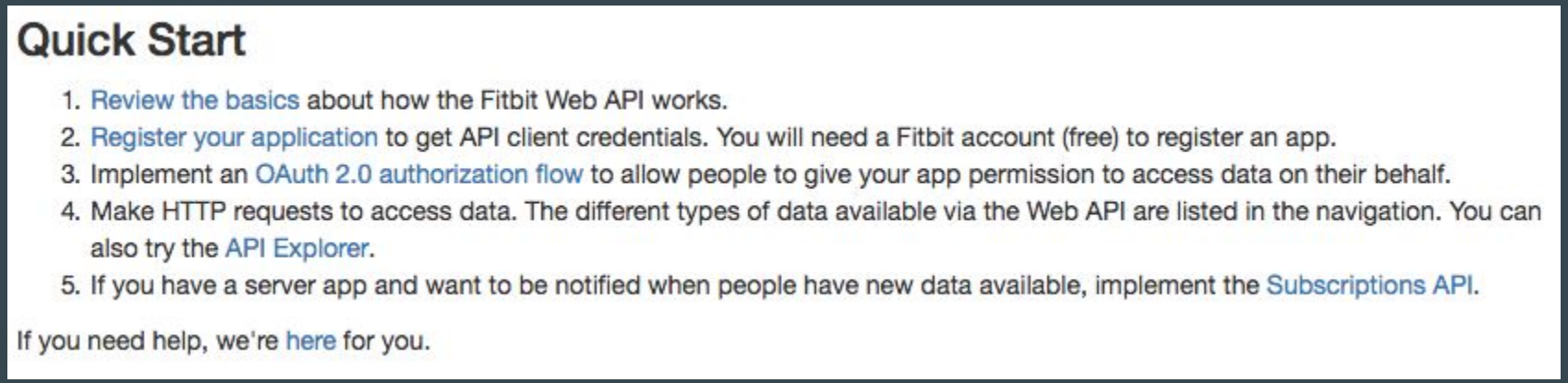
Fitbit

- 자신의 걸음, 운동량, 심박수, 수면패턴(자는시간) 등



Fitbit API

- Fitbit API : https://dev.fitbit.com/kr
- Fitbit API Document : https://dev.fitbit.com/docs/
  - Fitbit API가 어떻게 동작하는지 자세하게 나와있음



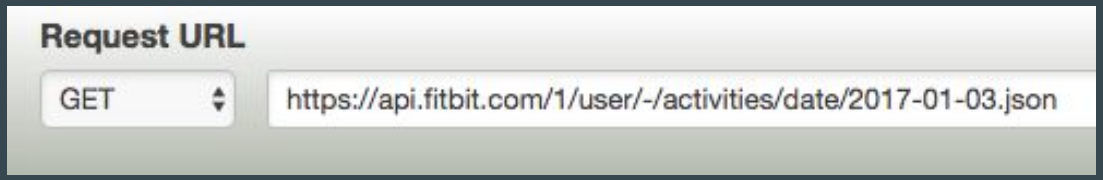
Quick Start

1. Review the basics about how the Fitbit Web API works.
2. Register your application to get API client credentials. You will need a Fitbit account (free) to register an app.
3. Implement an OAuth 2.0 authorization flow to allow people to give your app permission to access data on their behalf.
4. Make HTTP requests to access data. The different types of data available via the Web API are listed in the navigation. You can also try the API Explorer.
5. If you have a server app and want to be notified when people have new data available, implement the Subscriptions API.

If you need help, we're here for you.

Fitbit Restful API

- GET 메세지를 날려 응답을 받음




```
"floors": 15,
"heartRateZones": [
  {
    "caloriesOut": 2277.2673,
    "max": 98,
    "min": 30,
    "minutes": 1188,
    "name": "구간 밖"
  },
  {
    "caloriesOut": 891.429,
    "max": 137,
    "min": 98,
    "minutes": 158,
    "name": "지방 연소"
  },
  {
    "caloriesOut": 181.6892,
    "max": 166,
    "min": 137,
    "minutes": 15,
    "name": "심장강화운동"
  },
  {
    "caloriesOut": 0,
    "max": 220,
    "min": 166,
    "minutes": 0,
    "name": "정점"
  }
],
"lightlyActiveMinutes": 252,
"marginalCalories": 1128,
"restingHeartRate": 69,
"sedentaryMinutes": 695,
"steps": 13350,
"useEstimation": true,
"veryActiveMinutes": 49
}
```

Python Fitbit API 다운로드

python-fitbit

build passing | coverage 93% | requirements outdated

Fitbit API Python Client Implementation

For documentation: http://python-fitbit.readthedocs.org/

Requirements

- Python 2.7+
- python-dateutil (always)
- requests-oauthlib (always)
- Sphinx (to create the documention)
- tox (for running the tests)
- coverage (to create test coverage reports)

To use the library, you need to install the run time requirements:

```
sudo pip install -r requirements/base.txt
```

To modify and test the library, you need to install the developer requirements:

```
sudo pip install -r requirements/dev.txt
```

To run the library on a continuous integration server, you need to install the test requirements:

```
sudo pip install -r requirements/test.txt
```

Fitbit API using Python

- Python을 사용해 Fitbit API를 이용할 수 있음
- https://github.com/orcasgit/python-fitbit

Overview

This is a complete python implementation of the Fitbit API.

It uses oAuth for authentication, it supports both us and si measurements

Quickstart

If you are only retrieving data that doesn't require authorization, then you can use the unauthorized interface:

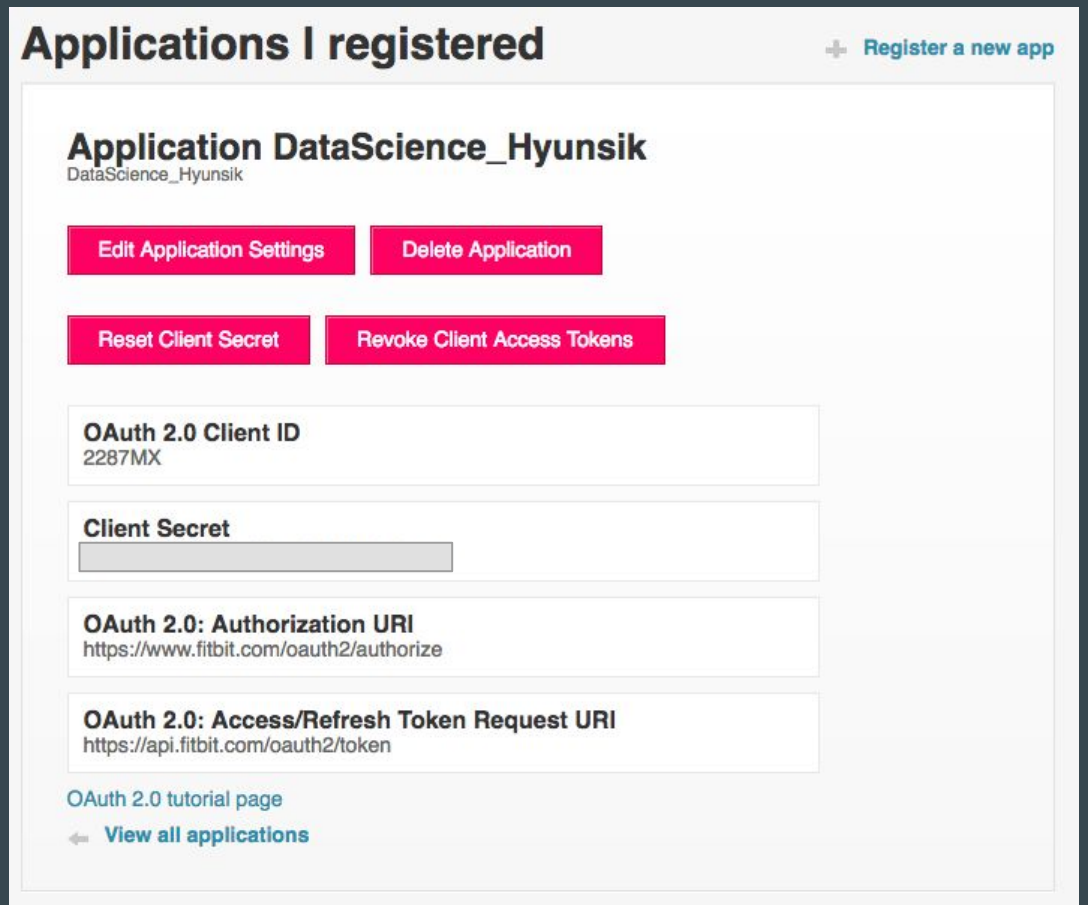
```
import fitbit
unauth_client = fitbit.Fitbit('<consumer_key>', '<consumer_secret>')
# certain methods do not require user keys
unauth_client.food_units()
```

Here is an example of authorizing with OAuth 2.0:

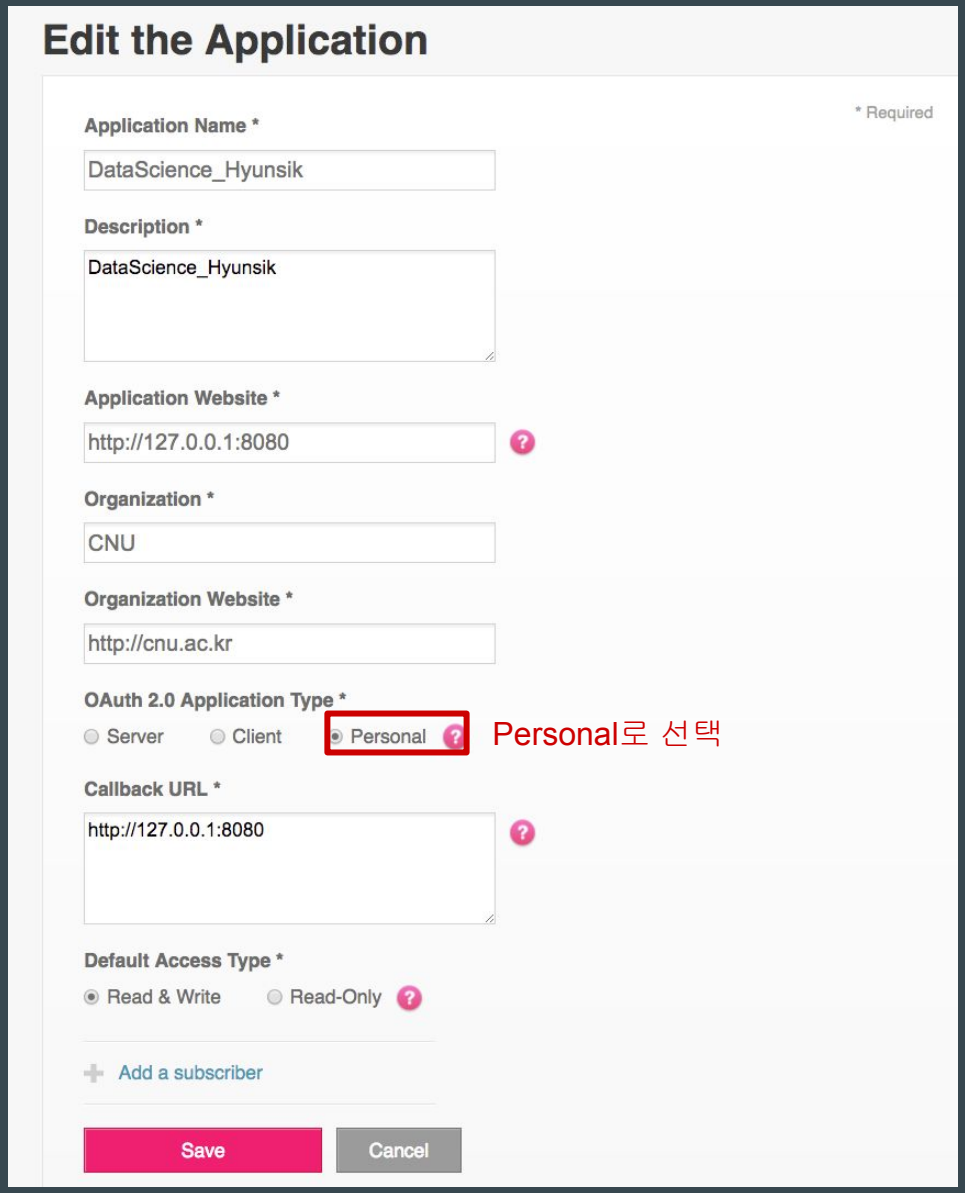
```
# You'll have to gather the tokens on your own, or use
# ./gather_keys_oauth2.py
authd_client = fitbit.Fitbit('<consumer_key>', '<consumer_secret>',
                             access_token='<access_token>', refresh_t
authd_client.sleep()
```

Fitbit API using Python 사용 조건

- Client App 등록
- https://dev.fitbit.com/apps



Fitbit API using Python 사용 조건

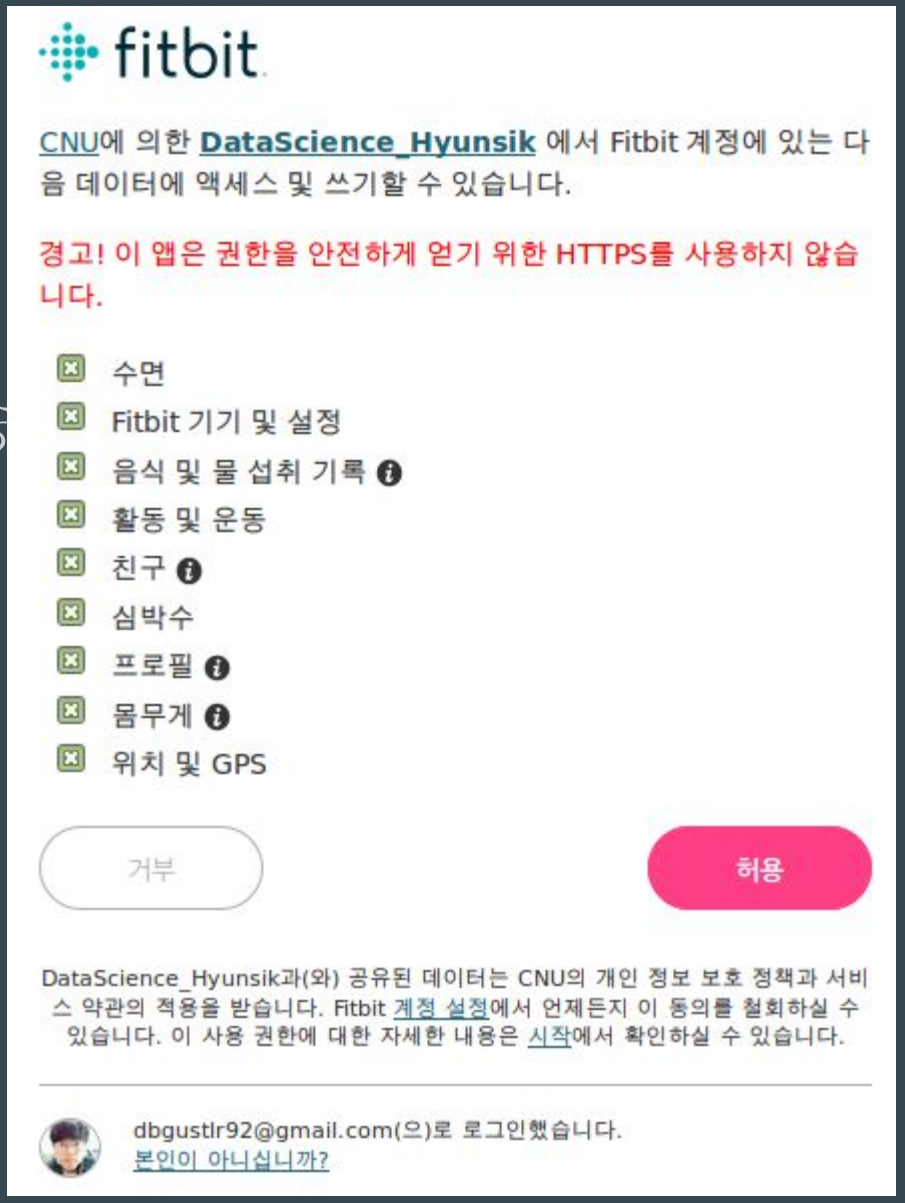


Python Fitbit API 으로 데이터 받기

1. Python 설치
2. Python Fitbit API 설치
3. O-Auth 2.0 script 실행
    a. Access token, Refresh token 받기
4. Fitbit API를 사용해 데이터 다운로드

Token 받기 (O-Auth 2.0)

- gather_keys_oauth2.py
- 실행시키고 뜨는 브라우저 승
- 승인 후 토큰 복사



Fitbit Data 받기 (with Python)

- 걸음수 받기 예제

```
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='2017-01-01', detail_level='15min')
print(intraday_step)

f = open('step.json','w')
json.dump(intraday_step, f)

```

Fitbit Data 받기 (with Python)

- 걸음수 데이터 결과


```
1 {"activities-steps-intraday": {"datasetType": "minute", "datasetInterval": 15, "dataset": [{"value": 47, "time": "00:00:00"}, {"value": 0, "time":
  "00:15:00"}, {"value": 7, "time": "00:30:00"}, {"value": 29, "time": "00:45:00"}, {"value": 0, "time": "01:00:00"}, {"value": 0, "time": "01:15:0
  0"}, {"value": 0, "time": "01:30:00"}, {"value": 0, "time": "01:45:00"}, {"value": 0, "time": "02:00:00"}, {"value": 0, "time": "02:15:00"}, {"val
  ue": 44, "time": "02:30:00"}, {"value": 0, "time": "02:45:00"}, {"value": 0, "time": "03:00:00"}, {"value": 13, "time": "03:15:00"}, {"value": 0,
  "time": "03:30:00"}, {"value": 0, "time": "03:45:00"}, {"value": 0, "time": "04:00:00"}, {"value": 8, "time": "04:15:00"}, {"value": 0, "time": "0
  4:30:00"}, {"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": 5, "time": "05:45:00"}, {"value": 0, "time": "06:00:00"}, {"value": 11, "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": 13, "time": "08:30:00"}, {"value": 0, "time": "08:45
  :00"}, {"value": 12, "time": "09:00:00"}, {"value": 0, "time": "09:15:00"}, {"value": 0, "time": "09:30:00"}, {"value": 10, "time": "09:45:00"}, {
  "value": 56, "time": "10:00:00"}, {"value": 73, "time": "10:15:00"}, {"value": 16, "time": "10:30:00"}, {"value": 165, "time": "10:45:00"}, {"valu
  e": 92, "time": "11:00:00"}, {"value": 246, "time": "11:15:00"}, {"value": 8, "time": "11:30:00"}, {"value": 250, "time": "11:45:00"}, {"value": 1
  60, "time": "12:00:00"}, {"value": 20, "time": "12:15:00"}, {"value": 150, "time": "12:30:00"}, {"value": 10, "time": "12:45:00"}, {"value": 0, "t
  ime": "13:00:00"}, {"value": 0, "time": "13:15:00"}, {"value": 108, "time": "13:30:00"}, {"value": 317, "time": "13:45:00"}, {"value": 309, "time"
  : "14:00:00"}, {"value": 48, "time": "14:15:00"}, {"value": 234, "time": "14:30:00"}, {"value": 75, "time": "14:45:00"}, {"value": 137, "time": "1
  5:00:00"}, {"value": 446, "time": "15:15:00"}, {"value": 310, "time": "15:30:00"}, {"value": 110, "time": "15:45:00"}, {"value": 774, "time": "16:
  00:00"}, {"value": 34, "time": "16:15:00"}, {"value": 184, "time": "16:30:00"}, {"value": 28, "time": "16:45:00"}, {"value": 0, "time": "17:00:00"
  }, {"value": 52, "time": "17:15:00"}, {"value": 92, "time": "17:30:00"}, {"value": 39, "time": "17:45:00"}, {"value": 12, "time": "18:00:00"}, {"v
  alue": 43, "time": "18:15:00"}, {"value": 55, "time": "18:30:00"}, {"value": 19, "time": "18:45:00"}, {"value": 45, "time": "19:00:00"}, {"value":
   30, "time": "19:15:00"}, {"value": 19, "time": "19:30:00"}, {"value": 23, "time": "19:45:00"}, {"value": 0, "time": "20:00:00"}, {"value": 0, "ti
  me": "20:15:00"}, {"value": 18, "time": "20:30:00"}, {"value": 0, "time": "20:45:00"}, {"value": 19, "time": "21:00:00"}, {"value": 0, "time": "21
  :15:00"}, {"value": 54, "time": "21:30:00"}, {"value": 57, "time": "21:45:00"}, {"value": 13, "time": "22:00:00"}, {"value": 0, "time": "22:15:00"
  }, {"value": 6, "time": "22:30:00"}, {"value": 0, "time": "22:45:00"}, {"value": 0, "time": "23:00:00"}, {"value": 0, "time": "23:15:00"}, {"value
  ": 32, "time": "23:30:00"}, {"value": 8, "time": "23:45:00"}]}, "activities-steps": [{"value": "5195", "dateTime": "2017-01-01"}]}
```

Fitbit Data 받기 (with Python)

- Heart rate 데이터 받기 예제
- http://janliphardt.com/2015/06/14/fitbit-api-and-high-resolution-heart-rate-data/

과제

- Fitbit O-Auth 2.0
  - O-Auth 2.0을 이용해 사용자 인증 후 데이터 받기
    - intraday
    - 2주동안의 걸음수
  - 다운로드한 데이터를 사용해 그래프 작성
  - 인증하는 과정과 데이터 받는방법을 보고서로 작성
    - 보고서에 코드 스크린샷과 설명

과제 제출 방법

- 과제 제출 기한 : **2017년 3월 29일 오후 6시까지!**
- Google Classroom에 제출!
  - **제출마감이후부터 24시간 경과시마다 만점의 20%씩 추가감점**
  - **예) 10점만점에 24시간 경과시 8점만점, 48시간 경과시 6점만점.., 5일경과시 제출점수 1점만 있음**
- 파일 제목 : **DS_학번_이름_주차.pdf, DS_학번_이름_주차.zip**
  - 보고서(PDF형태) : **HWP, DOC일경우 채점 안함**
  - 코드(zip형태)**(이번주 코드 제출 없음)**
  - **코드zip과 보고서를 하나로 압축하지 말 것!**
  - **파일 제목 및 형태 틀리면 -1점**

질문 사항

- 방문
  - 606호 (데이터네트워크 연구실)
- 메일
  - dbgustlr92@cs-cnu.org
- Google Class Room
  - Good!