# 205229118

# Mahalakshmi S

## Lab 8. Retrieving the user’s LinkedIn Profile and analysing the profile’s connections

In this lab, you need to access your LinkedIn profile and get the connections and perform some pre-processing steps on the data

1. Access the LinkedIn API and create an app and retrieve the API key and Secret key through the "Developer" section of your account settings by navigating directly to https://www.linkedin.com/secure/developer (https://www.linkedin.com/secure/developer).

### 2. Use LinkedIn OAuth credentials to receive an access token suitable for development and access your own data

In [3]:

```
import requests
import string
import random
CLIENT_ID = "78icgiayp0mu5z"
CLIENT_SECRET = "Pa8inm0abOmALXwb"
REDIRECT_URI = "http://localhost"

# Generate a random string to protect against cross-site request forgery
letters = string.ascii_lowercase
CSRF_TOKEN = ''.join(random.choice(letters) for i in range(24))

auth_params = {'response_type': 'code',
               'client_id': CLIENT_ID,
               'redirect_uri': REDIRECT_URI,
               'state': CSRF_TOKEN,
               'scope': 'r_liteprofile,r_emailaddress,w_member_social'}

html = requests.get("https://www.linkedin.com/oauth/v2/authorization",
                    params = auth_params)

# Print the link to the approval page
print(html.url)
```

```
https://www.linkedin.com/login?session_redirect=%2Foauth%2Fv2%2Flogin-succes
s%3Fapp_id%3D113884024%26auth_type%3DAC%26flow%3D%257B%2522authorizationTyp
e%2522%253A%2522OAUTH2_AUTHORIZATION_CODE%2522%252C%2522redirectUri%2522%253
A%2522http%253A%252F%252Flocalhost%2522%252C%2522currentStage%2522%253A%2522
LOGIN_SUCCESS%2522%252C%2522currentSubStage%2522%253A0%252C%2522authFlowNam
e%2522%253A%2522generic-permission-list%2522%252C%2522appId%2522%253A1138840
24%252C%2522creationTime%2522%253A1631774509463%252C%2522state%2522%253A%252
2qtjjmymklcgtwrdjpfrylnrv%2522%252C%2522scope%2522%253A%2522r_liteprofile%25
2Cr_emailaddress%252Cw_member_social%2522%257D&fromSignIn=1&trk=oauth&cancel
_redirect=%2Foauth%2Fv2%2Flogin-cancel%3Fapp_id%3D113884024%26auth_type%3DA
C%26flow%3D%257B%2522authorizationType%2522%253A%2522OAUTH2_AUTHORIZATION_CO
DE%2522%252C%2522redirectUri%2522%253A%2522http%253A%252F%252Flocalhost%252
2%252C%2522currentStage%2522%253A%2522LOGIN_SUCCESS%2522%252C%2522currentSub
Stage%2522%253A0%252C%2522authFlowName%2522%253A%2522generic-permission-lis
t%2522%252C%2522appId%2522%253A113884024%252C%2522creationTime%2522%253A1631
774509463%252C%2522state%2522%253A%2522qtjjmymklcgtwrdjpfrylnrv%2522%252C%25
22scope%2522%253A%2522r_liteprofile%252Cr_emailaddress%252Cw_member_social%2
522%257D (https://www.linkedin.com/login?session_redirect=%2Foauth%2Fv2%2Flo
gin-success%3Fapp_id%3D113884024%26auth_type%3DAC%26flow%3D%257B%2522authori
zationType%2522%253A%2522OAUTH2_AUTHORIZATION_CODE%2522%252C%2522redirectUr
i%2522%253A%2522http%253A%252F%252Flocalhost%2522%252C%2522currentStage%252
2%253A%2522LOGIN_SUCCESS%2522%252C%2522currentSubStage%2522%253A0%252C%2522a
uthFlowName%2522%253A%2522generic-permission-list%2522%252C%2522appId%2522%2
53A113884024%252C%2522creationTime%2522%253A1631774509463%252C%2522state%252
2%253A%2522qtjjmymklcgtwrdjpfrylnrv%2522%252C%2522scope%2522%253A%2522r_lite
profile%252Cr_emailaddress%252Cw_member_social%2522%257D&fromSignIn=1&trk=oa
uth&cancel_redirect=%2Foauth%2Fv2%2Flogin-cancel%3Fapp_id%3D113884024%26auth
_type%3DAC%26flow%3D%257B%2522authorizationType%2522%253A%2522OAUTH2_AUTHORI
ZATION_CODE%2522%252C%2522redirectUri%2522%253A%2522http%253A%252F%252Flocal
host%2522%252C%2522currentStage%2522%253A%2522LOGIN_SUCCESS%2522%252C%2522cu
rrentSubStage%2522%253A0%252C%2522authFlowName%2522%253A%2522generic-permiss
ion-list%2522%252C%2522appId%2522%253A113884024%252C%2522creationTime%2522%2
53A1631774509463%252C%2522state%2522%253A%2522qtjjmymklcgtwrdjpfrylnrv%2522%
252C%2522scope%2522%253A%2522r_liteprofile%252Cr_emailaddress%252Cw_member_s
ocial%2522%257D)
```

### 3. Inspect the address bar of your browser once you reach your redirect page and Copy the code after '&code=...', but don't include '&state=...' and paste it in the code below

In [7]:

```
NuFP5agaJM6ZYNcUiujDoR9FwGDepsa_60g-Usnlz9ImorxEQEjD7WR2mXJxKFZZEAJXKga75ECnbe3uUGDXu2zPGCb
```

```
Access Token: AQXOPmVPhZtiQvaKJpWqx_MQysPS-c9R8UsNO1x0GzIW-YZoBkE5e3ZVmR5eGy
wexH7S9OzAeFdDgTCkBtqeZLxM5QcG-gzcGpq62vEGmsd1Jtt9P8RLLPKXA1vWJQR8Yh6VyZoNfP
59_5oJOXngbpRfbnHQLdKXp3cm0tIHnPUJlmAYDESF6lOPuXh1xTb5ynww_vkPygRk6kBzaAcv9W
dnP-HT_n41zGwA6jN4zOKTpCZJCrnEheZ_QZBVuv7bUktp4ftAIKq-5ByP5qPApnER7GIHQ9aFlw
aKkfEMqAguxdW5TtXhmXTb4qyV0LW5RyDeSoiRCXno8D-tnfSIhuC8ky3mtA
Expires in (seconds): 5183999
```

### 4. Make a HTTP request to access personal profile

In [8]:

```
import json
params = {'oauth2_access_token': access_token,
          'fields': ["localizedFirstName,localizedLastName,id"]}
response = requests.get('https://api.linkedin.com/v2/me', params = params)
print(json.dumps(response.json(), indent=1))
```

```
{
 "localizedLastName": "S",
 "id": "VEYlKD4E-7",
 "localizedFirstName": "Mahalakshmi"
}
```

### 5. Download your LinkedIn profile data and read the connections data as a CSV file from the URL [https://www.linkedin.com/psettings/member-data (https://www.linkedin.com/psettings/member-data)](https://www.linkedin.com/psettings/member-data)

In [17]:

```
import os
import csv

# Point this to your 'Connections.csv' file.
CSV_FILE = os.path.join('Connections.csv')

csvReader = csv.DictReader(open(CSV_FILE), delimiter=',', quotechar='"')
contacts = [row for row in csvReader]
```

In [18]:

```
contacts
```

```
              ('Email Address', ''),
              ('Company', 'The Sparks Foundation'),
              ('Position', 'Data science and business analytics intern'),
              ('Connected On', '6-Jun-21')]),
 OrderedDict([('First Name', 'Mohamed'),
              ('Last Name', 'Sahim'),
              ('Email Address', ''),
              ('Company', 'The Sparks Foundation'),
              ('Position', 'Data Science and Business Analytics'),
              ('Connected On', '5-Jun-21')]),
 OrderedDict([('First Name', 'Jayasurya'),
              ('Last Name', 'V'),
              ('Email Address', ''),
              ('Company', 'Actify Data Labs'),
              ('Position', 'Data Analyst'),
              ('Connected On', '5-Jun-21')]),
 OrderedDict([('First Name', 'Vissweswaran'),
              ('Last Name', 'C'),
              ('Email Address', ''),
              ('Company', 'The Sparks Foundation'),
```

## 6. Apply some transformations to the connections dataset and retrieve the following:

a. Find the list of associated organisations and get the frequency b. Find the list of professional titles and the associated frequency

In [20]:

```
from prettytable import PrettyTable # pip install prettytable
from collections import Counter
from operator import itemgetter

# Define a set of transforms that converts the first item
# to the second item. Here, we're simply handling some
# commonly known abbreviations, stripping off common suffixes,
# etc.

transforms = [(', Inc.', ''), (', Inc', ''), (', LLC', ''), (', LLP', ''),
              (' LLC', ''), (' Inc.', ''), (' Inc', '')]

companies = [c['Company'].strip() for c in contacts if c['Company'].strip() != '']

for i, _ in enumerate(companies):
    for transform in transforms:
        companies[i] = companies[i].replace(*transform)

pt = PrettyTable(field_names=['Company', 'Freq'])
pt.align = 'l'
c = Counter(companies)

[pt.add_row([company, freq]) for (company, freq) in sorted(c.items(), key=itemgetter(1), re

print(pt)
```

```
+-------------------------------------------------+------+
| Company                                         | Freq |
+-------------------------------------------------+------+
| The Sparks Foundation                           | 11   |
| Bishop Heber College, Tiruchirappalli - 620 017. | 4    |
+-------------------------------------------------+------+
```

In [21]:

```
transforms = [
    ('Sr.', 'Senior'),
    ('Sr', 'Senior'),
    ('Jr.', 'Junior'),
    ('Jr', 'Junior'),
    ('CEO', 'Chief Executive Officer'),
    ('COO', 'Chief Operating Officer'),
    ('CTO', 'Chief Technology Officer'),
    ('CFO', 'Chief Finance Officer'),
    ('VP', 'Vice President'),
    ]

# Read in a list of titles and split apart
# any combined titles like "President/CEO."
# Other variations could be handled as well, such
# as "President & CEO", "President and CEO", etc.

titles = []
for contact in contacts:
    titles.extend([t.strip() for t in contact['Position'].split('/')
                   if contact['Position'].strip() != ''])

# Replace common/known abbreviations

for i, _ in enumerate(titles):
    for transform in transforms:
        titles[i] = titles[i].replace(*transform)

# Print out a table of titles sorted by frequency

pt = PrettyTable(field_names=['Job Title', 'Freq'])
pt.align = 'l'
c = Counter(titles)
[pt.add_row([title, freq])
 for (title, freq) in sorted(c.items(), key=itemgetter(1), reverse=True)
     if freq > 1]
print(pt)

# Print out a table of tokens sorted by frequency

tokens = []
for title in titles:
    tokens.extend([t.strip(',') for t in title.split()])
pt = PrettyTable(field_names=['Token', 'Freq'])
pt.align = 'l'
c = Counter(tokens)
[pt.add_row([token, freq])
 for (token, freq) in sorted(c.items(), key=itemgetter(1), reverse=True)
     if freq > 1 and len(token) > 2]
print(pt)
```

```
+-------------------------------------------+------+
| Job Title                                 | Freq |
+-------------------------------------------+------+
| Data Science and Business Analytics Intern  | 2    |
| Postgraduate Student                        | 2    |
| Data Science and Business Analystics Intern | 2    |
+-------------------------------------------+------+
```

| Token | Freq |
|---|---|
| Data | 16 |
| Science | 12 |
| and | 10 |
| Business | 8 |
| Intern | 6 |
| Analytics | 5 |
| Student | 3 |
| analytics | 3 |
| intern | 3 |
| Human | 2 |
| Resources | 2 |
| Executive | 2 |
| Trainee | 2 |
| Postgraduate | 2 |
| business | 2 |
| Analystics | 2 |