

225229110

HARI PRASATH S

Lab 3: Getting the stargazers of a GitHub Repository and create a network**Step-1: Install PyGithub requests and get information about your Github Profile**

In [1]:

```
import requests
from pprint import pprint
username = 'Harihp2002'
url = f'https://api.github.com/users/{username}'
user_data = requests.get(url).json()
pprint(user_data)

{'avatar_url': 'https://avatars.githubusercontent.com/u/138384616?v=4',
 'bio': None,
 'blog': '',
 'company': None,
 'created_at': '2023-07-02T15:06:53Z',
 'email': None,
 'events_url': 'https://api.github.com/users/Harihp2002/events{/privacy}',
 'followers': 0,
 'followers_url': 'https://api.github.com/users/Harihp2002/followers',
 'following': 0,
 'following_url': 'https://api.github.com/users/Harihp2002/following{/other_user}',
 'gists_url': 'https://api.github.com/users/Harihp2002/gists{/gist_id}',
 'gravatar_id': '',
 'hireable': None,
 'html_url': 'https://github.com/Harihp2002',
 'id': 138384616,
 'location': None,
 'login': 'Harihp2002',
 'name': None,
 'node_id': 'U_kgDOCD-U6A',
 'organizations_url': 'https://api.github.com/users/Harihp2002/orgs',
 'public_gists': 0,
 'public_repos': 4,
 'received_events_url': 'https://api.github.com/users/Harihp2002/received_events',
 'repos_url': 'https://api.github.com/users/Harihp2002/repos',
 'site_admin': False,
 'starred_url': 'https://api.github.com/users/Harihp2002/starred{/owner}/{repo}',
 'subscriptions_url': 'https://api.github.com/users/Harihp2002/subscriptions',
 'twitter_username': None,
 'type': 'User',
 'updated_at': '2023-07-20T03:32:33Z',
 'url': 'https://api.github.com/users/Harihp2002'}
```

Step-2: Getting Public repositories of a user

In [2]:

```
!pip install pygithub
```

Collecting pygithub

Downloading PyGithub-1.59.0-py3-none-any.whl (342 kB)

Requirement already satisfied: requests>=2.14.0 in c:\users\lenovo\anaconda3\lib\site-packages (from pygithub) (2.25.1)

Collecting pyjwt[crypto]>=2.4.0

Downloading PyJWT-2.8.0-py3-none-any.whl (22 kB)

Requirement already satisfied: pynacl>=1.4.0 in c:\users\lenovo\anaconda3\lib\site-packages (from pygithub) (1.4.0)

Collecting deprecated

Downloading Deprecated-1.2.14-py2.py3-none-any.whl (9.6 kB)

Requirement already satisfied: cryptography>=3.4.0 in c:\users\lenovo\anaconda3\lib\site-packages (from pyjwt[crypto]>=2.4.0->pygithub) (3.4.7)

Requirement already satisfied: cffi>=1.12 in c:\users\lenovo\anaconda3\lib\site-packages (from cryptography>=3.4.0->pyjwt[crypto]>=2.4.0->pygithub) (1.14.5)

Requirement already satisfied: pycparser in c:\users\lenovo\anaconda3\lib\site-packages (from cffi>=1.12->cryptography>=3.4.0->pyjwt[crypto]>=2.4.0->pygithub) (2.20)

Requirement already satisfied: six in c:\users\lenovo\anaconda3\lib\site-packages (from pynacl>=1.4.0->pygithub) (1.15.0)

Requirement already satisfied: idna<3,>=2.5 in c:\users\lenovo\anaconda3\lib\site-packages (from requests>=2.14.0->pygithub) (2.10)

Requirement already satisfied: chardet<5,>=3.0.2 in c:\users\lenovo\anaconda3\lib\site-packages (from requests>=2.14.0->pygithub) (4.0.0)

Requirement already satisfied: certifi>=2017.4.17 in c:\users\lenovo\anaconda3\lib\site-packages (from requests>=2.14.0->pygithub) (2020.12.5)

Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\lenovo\anaconda3\lib\site-packages (from requests>=2.14.0->pygithub) (1.26.4)

Requirement already satisfied: wrapt<2,>=1.10 in c:\users\lenovo\anaconda3\lib\site-packages (from deprecated->pygithub) (1.12.1)

Installing collected packages: pyjwt, deprecated, pygithub

Successfully installed deprecated-1.2.14 pygithub-1.59.0 pyjwt-2.8.0

In [4]:

```
import base64
from github import Github
from pprint import pprint
username="Harihp2002"
g=Github()
user=g.get_user(username)
for repo in user.get_repos():
    print(repo)
```

Repository(full_name="Harihp2002/admission-flask")

Repository(full_name="Harihp2002/admission_pred")

Repository(full_name="Harihp2002/Admission_pred_PMLmicroproject")

Repository(full_name="Harihp2002/Predict-webapp")

In [5]:

```
from github import Github
ACCESS_TOKEN="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"
USER="ptwobrussell"
REPO="Mining-the-Social-Web"
#REPO="Mining-the-Social-Web-2nd-Edition"
client=Github(ACCESS_TOKEN, per_page=100)
user=client.get_user(USER)
repo=user.get_repo(REPO)
stargazers=[s for s in repo.get_stargazers()]
print("Number of stargazers", len(stargazers))
```

Number of stargazers 1210

Step-4: Constructing an ego graph of a repository and its stargazers

In [7]:

```
import networkx as nx
g=nx.DiGraph()
g.add_node(repo.name+"(repo)",type='repo',lang=repo.language,owner=user.login)
for sg in stargazers:
    g.add_node(sg.login+"(user)",type='user')
    g.add_edge(sg.login+"(user)",repo.name+"(repo)",type='gazes')
```

Step-5: Perform handy graph operations

In [8]:

```

print(nx.info(g))
print(g.nodes['Mining-the-Social-Web(repo)'])
print(g.nodes['ptwobrussell(user)'])
print(g['ptwobrussell(user)']['Mining-the-Social-Web(repo)'])
print(g['ptwobrussell(user)'])
print(g['Mining-the-Social-Web(repo)'])
print(g.in_edges(['ptwobrussell(user)']))
print(g.out_edges(['ptwobrussell(user)']))
print(g.in_edges(['Mining-the-Social-Web(repo)']))
print(g.out_edges(['Mining-the-Social-Web(repo)']))

```

Name:

Type: DiGraph

Number of nodes: 1211

Number of edges: 1210

Average in degree: 0.9992

Average out degree: 0.9992

{ 'type': 'repo', 'lang': 'JavaScript', 'owner': 'ptwobrussell' }

{ 'type': 'user' }

{ 'type': 'gazes' }

{ 'Mining-the-Social-Web(repo)': { 'type': 'gazes' } }

{ }

[]

[('ptwobrussell(user)', 'Mining-the-Social-Web(repo)')]

```

[( 'rdempsey(user)', 'Mining-the-Social-Web(repo)' ), ( 'prb(user)', 'Mining-the-Social-Web(repo)' ), ( 'mcroydon(user)', 'Mining-the-Social-Web(repo)' ), ( 'twleung(user)', 'Mining-the-Social-Web(repo)' ), ( 'kevinchiu(user)', 'Mining-the-Social-Web(repo)' ), ( 'nikolay(user)', 'Mining-the-Social-Web(repo)' ), ( 'tswicegood(user)', 'Mining-the-Social-Web(repo)' ), ( 'ngpestelos(user)', 'Mining-the-Social-Web(repo)' ), ( 'darron(user)', 'Mining-the-Social-Web(repo)' )]

```

In []: