In [1]:

```
#!/usr/bin/env python
# encoding=utf-8
from __future__ import print_function

import os
import requests
import re
import time
import xml.dom.minidom
import json
import sys
import math
import subprocess
import ssl
import threading
import urllib
import urllib.request
import pandas as pd


DEBUG = False

MAX_GROUP_NUM = 2  # members in each group
INTERFACE_CALLING_INTERVAL = 5  # the time interval between each call
MAX_PROGRESS_LEN = 50

QRImagePath = os.path.join(os.getcwd(), 'qrcode.jpg')

tip = 0
uuid = ''

base_uri = ''
redirect_uri = ''
push_uri = ''

skey = ''
wxsid = ''
wxuin = ''
pass_ticket = ''
deviceId = 'e000000000000000'

BaseRequest = {}

ContactList = []
My = []
SyncKey = []

try:
    xrange
    range = xrange
```

```
except:
    # python 3
    pass


def responseState(func, BaseResponse):
    ErrMsg = BaseResponse['ErrMsg']
    Ret = BaseResponse['Ret']
    if DEBUG or Ret != 0:
        print('func: %s, Ret: %d, ErrMsg: %s' % (func, Ret, ErrMsg))

    if Ret != 0:
        return False

    return True



def getUUID():
    global uuid

    url = 'https://login.weixin.qq.com/jslogin'
    params = {
        'appid': 'wx782c26e4c19acffb',
        'fun': 'new',
        'lang': 'zh_CN',
        '_': int(time.time()),
    }

    r= myRequests.get(url=url, params=params)
    r.encoding = 'utf-8'
    data = r.text

    # print(data)

    # window.QRLogin.code = 200; window.QRLogin.uuid = "oZwt_bFfRg==";
    regx = r'window.QRLogin.code = (\d+); window.QRLogin.uuid = "(\S+?)"'
    pm = re.search(regx, data)

    code = pm.group(1)
    uuid = pm.group(2)

    if code == '200':
        return True

    return False


def showQRImage():
    global tip

    url = 'https://login.weixin.qq.com/qrcode/' + uuid
    params = {
        't': 'webwx',
```

```
        '_': int(time.time()),
    }

    r = myRequests.get(url=url, params=params)

    tip = 1

    f = open(QRImagePath, 'wb')
    f.write(r.content)
    f.close()
    time.sleep(1)

    if sys.platform.find('darwin') >= 0:
        subprocess.call(['open', QRImagePath])
    else:
        subprocess.call(['xdg-open', QRImagePath])

    print('please use WeChat to scan QR code to log in..')


def waitForLogin():
    global tip, base_uri, redirect_uri, push_uri

    url = 'https://login.weixin.qq.com/cgi-bin/mmwebwx-bin/login?tip=%s&uuid=%s&_=%s
        tip, uuid, int(time.time()))

    r = myRequests.get(url=url)
    r.encoding = 'utf-8'
    data = r.text

    # print(data)

    # window.code=500;
    regx = r'window.code=(\d+);'
    pm = re.search(regx, data)

    code = pm.group(1)

    if code == '201':  # Already scanned
        print('scan successful, plsase tap on your phone to login in')
        tip = 0
    elif code == '200':  # Already logined
        print('now logging in ...')
        regx = r'window.redirect_uri="(\S+?)";'
        pm = re.search(regx, data)
        redirect_uri = pm.group(1) + '&fun=new'
        base_uri = redirect_uri[:redirect_uri.rfind('/')]

        services = [
            ('wx2.qq.com', 'webpush2.weixin.qq.com'),
            ('qq.com', 'webpush.weixin.qq.com'),
            ('web1.wechat.com', 'webpush1.wechat.com'),
            ('web2.wechat.com', 'webpush2.wechat.com'),
            ('wechat.com', 'webpush.wechat.com'),
```

```
            ('web1.wechatapp.com', 'webpush1.wechatapp.com'),
        ]
        push_uri = base_uri
        for (searchUrl, pushUrl) in services:
            if base_uri.find(searchUrl) >= 0:
                push_uri = 'https://%s/cgi-bin/mmwebwx-bin' % pushUrl
                break

        # closeQRImage
        if sys.platform.find('darwin') >= 0:  # for OSX with Preview
            os.system("osascript -e 'quit app \"Preview\"'")
    elif code == '408':  # overtime
        pass
    # elif code == '400' or code == '500':

    return code


def login():
    global skey, wxsid, wxuin, pass_ticket, BaseRequest

    r = myRequests.get(url=redirect_uri)
    r.encoding = 'utf-8'
    data = r.text

    # print(data)

    doc = xml.dom.minidom.parseString(data)
    root = doc.documentElement

    for node in root.childNodes:
        if node.nodeName == 'skey':
            skey = node.childNodes[0].data
        elif node.nodeName == 'wxsid':
            wxsid = node.childNodes[0].data
        elif node.nodeName == 'wxuin':
            wxuin = node.childNodes[0].data
        elif node.nodeName == 'pass_ticket':
            pass_ticket = node.childNodes[0].data

    # print('skey: %s, wxsid: %s, wxuin: %s, pass_ticket: %s' % (skey, wxsid,
    # wxuin, pass_ticket))

    if not all((skey, wxsid, wxuin, pass_ticket)):
        return False

    BaseRequest = {
        'Uin': int(wxuin),
        'Sid': wxsid,
        'Skey': skey,
        'DeviceID': deviceId,
    }

    return True
```

```
def webwxinit():

    url = (base_uri +
        '/webwxinit?pass_ticket=%s&skey=%s&r=%s' % (
            pass_ticket, skey, int(time.time())) )
    params  = {'BaseRequest': BaseRequest }
    headers = {'content-type': 'application/json; charset=UTF-8'}

    r = myRequests.post(url=url, data=json.dumps(params),headers=headers)
    r.encoding = 'utf-8'
    data = r.json()

    if DEBUG:
        f = open(os.path.join(os.getcwd(), 'webwxinit.json'), 'wb')
        f.write(r.content)
        f.close()


    # print(data)

    global ContactList, My, SyncKey
    dic = data
    ContactList = dic['ContactList']
    My = dic['User']
    SyncKey = dic['SyncKey']

    state = responseState('webwxinit', dic['BaseResponse'])
    return state


def webwxgetcontact():

    url = (base_uri +
        '/webwxgetcontact?pass_ticket=%s&skey=%s&r=%s' % (
            pass_ticket, skey, int(time.time())) )
    headers = {'content-type': 'application/json; charset=UTF-8'}


    r = myRequests.post(url=url,headers=headers)
    r.encoding = 'utf-8'
    data = r.json()

    if DEBUG:
        f = open(os.path.join(os.getcwd(), 'webwxgetcontact.json'), 'wb')
        f.write(r.content)
        f.close()


    dic = data
    MemberList = dic['MemberList']

    SpecialUsers = ["newsapp", "fmessage", "filehelper", "weibo", "qqmail", "tmessag
```

```
                    "meishiapp", "feedsapp", "voip", "blogappweixin", "weixin", "bra
    for i in range(len(MemberList) - 1, -1, -1):
        Member = MemberList[i]
        if Member['VerifyFlag'] & 8 != 0:  # public account
            MemberList.remove(Member)
        elif Member['UserName'] in SpecialUsers:  # special acount
            MemberList.remove(Member)
        elif Member['UserName'].find('@@') != -1:  # group chat
            MemberList.remove(Member)
        elif Member['UserName'] == My['UserName']:  # myself
            MemberList.remove(Member)

    return MemberList


def syncKey():
    SyncKeyItems = ['%s_%s' % (item['Key'], item['Val'])
                    for item in SyncKey['List']]
    SyncKeyStr = '|'.join(SyncKeyItems)
    return SyncKeyStr


def syncCheck():
    url = push_uri + '/synccheck?'
    params = {
        'skey': BaseRequest['Skey'],
        'sid': BaseRequest['Sid'],
        'uin': BaseRequest['Uin'],
        'deviceId': BaseRequest['DeviceID'],
        'synckey': syncKey(),
        'r': int(time.time()),
    }

    r = myRequests.get(url=url,params=params)
    r.encoding = 'utf-8'
    data = r.text

    # print(data)

    # window.synccheck={retcode:"0",selector:"2"}
    regx = r'window.synccheck={retcode:"(\d+)",selector:"(\d+)"}'
    pm = re.search(regx, data)

    retcode = pm.group(1)
    selector = pm.group(2)

    return selector


def webwxsync():
    global SyncKey

    url = base_uri + '/webwxsync?lang=zh_CN&skey=%s&sid=%s&pass_ticket=%s' % (
        BaseRequest['Skey'], BaseRequest['Sid'], urllib.quote_plus(pass_ticket))
```

```
    params = {
        'BaseRequest': BaseRequest,
        'SyncKey': SyncKey,
        'rr': ~int(time.time()),
    }
    headers = {'content-type': 'application/json; charset=UTF-8'}

    r = myRequests.post(url=url, data=json.dumps(params))
    r.encoding = 'utf-8'
    data = r.json()

    # print(data)

    dic = data
    SyncKey = dic['SyncKey']

    state = responseState('webwxsync', dic['BaseResponse'])
    return state


def heartBeatLoop():
    while True:
        selector = syncCheck()
        if selector != '0':
            webwxsync()
        time.sleep(1)


def main():
    global myRequests

    if hasattr(ssl, '_create_unverified_context'):
        ssl._create_default_https_context = ssl._create_unverified_context

    headers = {'User-agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_2) AppleW
    myRequests = requests.Session()
    myRequests.headers.update(headers)


    if not getUUID():
        print('Acquiring uuid fail...')
        return

    print('Requiring QR code Images ...')
    showQRImage()

    while waitForLogin() != '200':
        pass

    os.remove(QRImagePath)

    if not login():
        print('Login fail..')
        return
```

```
if not webwxinit():
    print('Initialization fail..')
    return

MemberList = webwxgetcontact()

threading.Thread(target=heartBeatLoop)

MemberCount = len(MemberList)
print('there are totally %s friends in your friend list' % MemberCount)

d = {}
imageIndex = 0
for Member in MemberList:
    imageIndex = imageIndex + 1
    if imageIndex == 10:
        break;
    name = '/Users/xiaosu/Documents/a2017Fall/project/image'+str(imageIndex)+'.j
    imageUrl = 'https://wx2.qq.com'+Member['HeadImgUrl']
    r = myRequests.get(url=imageUrl,headers=headers)
    imageContent = (r.content)
    fileImage = open(name,'wb')
    fileImage.write(imageContent)
    fileImage.close()
    print('Downloading: '+str(imageIndex)+'th friend picture')
    d[Member['UserName']] = (Member['NickName'], Member['RemarkName'])
        city = Member['City']
        city = 'nocity' if city == '' else  city
        name = Member['NickName']
        name = 'noname' if name == '' else  name
        sign = Member['Signature']
        sign = 'nosign' if sign == '' else  sign
        remark = Member['RemarkName']
        remark = 'noremark' if remark == '' else remark
        alias = Member['Alias']
        alias = 'noalias' if alias == '' else alias
        nick = Member['NickName']
        nick = 'nonick' if nick == '' else nick
        print(name,'  ^+*+^  ',city,'  ^+*+^  ',Member['Sex'],' ^+*+^ ',Member['Sta

    dic = {}
    mlist = MemberList[0].keys()
    for l in mlist:
        dic[l] = [r[l] for r in MemberList]
        DataFrame=pd.DataFrame(dic).to_csv('/Users/xiaosu/Documents/a2017Fall/projec

if __name__ == '__main__':

    main()
    print('press enter to exit...')
    input()
```

```
Requiring QR code Images ...
please use WeChat to scan QR code to log in..
scan successful, plsase tap on your phone to login in
now logging in ...
there are totally 370 friends in your friend list
Downloading: 1th friend picture
ADAOUD  ^+*+^  nocity  ^+*+^  0  ^+*+^  1  ^+*+^  nosign  ^+*+^  A
DAOUD David  ^+*+^  noalias  ^+*+^  ADAOUD
Downloading: 2th friend picture
2000  ^+*+^  武汉  ^+*+^  2  ^+*+^  0  ^+*+^  nosign  ^+*+^  九九妹
妹  ^+*+^  noalias  ^+*+^  2000
Downloading: 3th friend picture
Gavin  ^+*+^  nocity  ^+*+^  0  ^+*+^  0  ^+*+^  nosign  ^+*+^  no
remark  ^+*+^  noalias  ^+*+^  Gavin
Downloading: 4th friend picture
不会哭的阿狸  ^+*+^  武汉  ^+*+^  2  ^+*+^  0  ^+*+^  天气不错，心情很好
^+*+^  贾艳丽/M  ^+*+^  noalias  ^+*+^  不会哭的阿狸
Downloading: 5th friend picture
现实狠无奈-_-  ^+*+^  荆州  ^+*+^  1  ^+*+^  0  ^+*+^  压力山大啊！  ^+
*+^  周全/1/M  ^+*+^  noalias  ^+*+^  现实狠无奈-_-
Downloading: 6th friend picture
Adam Dresher  ^+*+^  佛山  ^+*+^  1  ^+*+^  0  ^+*+^  doc?  ^+*+^
noremark  ^+*+^  noalias  ^+*+^  Adam Dresher
Downloading: 7th friend picture
王业祯  ^+*+^  长宁  ^+*+^  1  ^+*+^  0  ^+*+^  work, don't play  ^+
*+^  noremark  ^+*+^  noalias  ^+*+^  王业祯
Downloading: 8th friend picture
ALI  ^+*+^  nocity  ^+*+^  0  ^+*+^  0  ^+*+^  nosign  ^+*+^  nore
mark  ^+*+^  noalias  ^+*+^  ALI
Downloading: 9th friend picture
刘杰锋  ^+*+^  广州  ^+*+^  1  ^+*+^  0  ^+*+^  nosign  ^+*+^  norem
ark  ^+*+^  noalias  ^+*+^  刘杰锋
press enter to exit...
```

In [4]:

```
#-*- coding: UTF-8 -*-
import pandas as pd
df = pd.read_csv('/Users/xiaosu/Documents/a2017Fall/project/contact.csv',encoding='u

#Calculate the friend number in different city
def city():
    address = df['City'].value_counts()
    print (address)

def InfoCalculate():
    name = df['UserName']

    remarkCount = 0
    maleCount = 0
    femaleCount = 0
    for i in range(1,len(name)):
```

```
        if judgeGender(i) == 'male':
            maleCount = maleCount + 1
        elif judgeGender(i) == 'female':
            femaleCount = femaleCount + 1
    print ('WeChat Total Friend number: ',str(len(name)))
    print ('total male number: ',maleCount,'total female number: ',femaleCount)
    print ('\n','Calculate number of friends in different city' )
    city()

def judgeGender(index):
    gender = df['Sex']
    if gender[index] == 1:
        return ('male')
    elif gender[index] == 2:
        return ('female')
    else:
        return ('unknown')

if __name__=='__main__':
    InfoCalculate()
```

```
WeChat Total Friend number:  370
total male number:  163 total female number:  176

 Calculate number of friends in different city
New York City         43
武汉                    38
深圳                    35
广州                    33
New York              16
海淀                    10
纽约                     8
长沙                     6
咸宁                     4
佛山                     3
杭州                     3
东城                     3
San Francisco         3
江门                     2
Santa Clara City      2
杨浦                     2
SantaClara            2
London                2
成都                     2
朝阳                     2
San Francisco City    2
苏州                     2
Cupertino City        1
长宁                     1
泰安                     1
吐鲁番                    1
中山                     1
绍兴                     1
台东县                    1
```

```
Vancouver                1
                        ..
南开                       1
鞍山                       1
河源                       1
Ithaca                   1
汕头                       1
大堂区                      1
浦东新区                     1
安庆                       1
Evanston City            1
三亚                       1
合肥                       1
Rochester City           1
Seattle                  1
河北                       1
海口                       1
锦州                       1
Santa Clara County       1
Gold Coast               1
Seattle City             1
韶关                       1
Jersey City              1
开封                       1
台北市                      1
Rochester                1
黄石                       1
Santa Barbara City       1
普陀                        1
Los Angeles              1
Palo Alto                1
Others                   1
Name: City, Length: 79, dtype: int64
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: