

GAB USER

Steps to Run .py File:

step 1 create account on gab

step2 get one gab_user_name which you want to extract info

step3: after getting 3 inputs 1. username 2.password 3.gab_user_name

step4 feed this info in the user_info function and run the .py file

How I have written the py file

- I used python selenium and used CSS selector or tag name to get the elements
- first login
- extract user info 1. gab 2. likes 3.repost 4.date joined 5.username
- extract last 50 posts using infinite scrolling extraction concept
- code is properly commented out you can find the py file and code in this document also

Code:

Login Function

```
def login(  
    username,  
    passwd,  
):  
    "login function!"  
    url = 'https://gab.com/auth/sign_in'  
    driver.get(url)  
    driver.implicitly_wait(10)  
    input_elements = driver.find_elements_by_tag_name('input')  
    button = driver.find_element_by_tag_name('button')  
    email = input_elements[1]  
    password = input_elements[2]  
    email.send_keys(username)  
    password.send_keys(passwd)  
    button.click()  
    time.sleep(2)
```

UserInfo function

```
def user_info(
    gab_user_name = "impuneet",
    username = 'impuneet',
    passwd = 'Punit1811@'
):

    #login into window
    login(
        username=username,
        passwd=passwd,
    )

    #open user info in another browser
    user_info = {}
    driver.execute_script("window.open('\" + \"https://gab.com/\"+ gab_user_name + '\" );")
    driver.switch_to.window(driver.window_handles[1])
    time.sleep(3)

    #user_name or datejoined
    user_info['username'] = gab_user_name
    date_box = driver.find_element_by_css_selector('._UuSG._ayWa._3dGg1.Vlb1o._1vyTb')
    driver.execute_script("arguments[0].scrollIntoView();",date_box)
    join_element = date_box.find_elements_by_tag_name('span')
    user_info['date_joined'] = join_element[8].text

    #user_Basi._info
    first_box = driver.find_element_by_css_selector('._UuSG.w77Za')
    infos = first_box.find_elements_by_tag_name('a')
    user_info['gab'] = infos[12].text
    user_info['followers'] = infos[13].text
    user_info['following'] = infos[14].text

    #last 50 post retries
    timeline = driver.find_element_by_css_selector('._UuSG.w77Za._3cqkW')
    articles = timeline.find_elements_by_tag_name('article')
    driver.execute_script("arguments[0].scrollIntoView();", articles[len(articles)-1])
    comment_extract(
        timeline,
        articles
    )
```

```
timeline = driver.find_element_by_css_selector('._UuSG.w77Za._3cqkW')
articles = timeline.find_elements_by_tag_name('article')
post_info = last_50_post_info_extract(articles)
user_info['last_50_post'] = post_info
```

```
return user_info
```

Last 50 post extract[Infinite Scrolling]

```
def comment_extract(
    timeline,
    articles
):
    current_element = 0
    while True:
        if current_element > 50:
            print(len(articles))
            time.sleep(4)
            break
        else:
            articles = timeline.find_elements_by_tag_name('article')
            if current_element == len(articles):
                break
            driver.execute_script("arguments[0].scrollIntoView();", articles[len(articles)-1])
            print(len(articles))
            current_element = len(articles)
            time.sleep(4)
```

```
def last_50_post_info_extract(
    comments,
):
    comment_info = {}
    for idx, comment in enumerate(comments):
        comment_info[idx] = {}

        driver.execute_script("arguments[0].scrollIntoView();", comment)
        text = comment.find_element_by_css_selector('._UuSG')
        box_extra = comment.find_elements_by_tag_name('button')

    try:
```

```
comts_num = text.find_elements_by_tag_name('a')[4].text
except:
    comts_num = 0

#save result in dict
likes = box_extra[1].text
reposts = box_extra[2].text
comment_info[idx]['text'] = text.text
comment_info[idx]['likes'] = likes
comment_info[idx]['no_of_comments'] = comts_num
comment_info[idx]['reposts'] = reposts
return comment_info
```