

Extract_LinkedIn_articles_using_selenium-and-BeautifulSoup-Python-

<> Code

! Issues

🔗 Pull requests

▶ Actions

📁 Projects

🛡 Security

📄 Insights

Join GitHub today

Dismiss

GitHub is home to over 50 million developers working together to host and review code, manage projects, and build software together.

Sign up

🔗 master ▾

⋮

Extract_LinkedIn_articles_using_selenium-and-BeautifulSoup-Python- / main.py



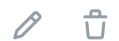
ENIGMA-exe Add files via upload

🕒 History

👤 1 contributor

Raw

Blame



143 lines (110 sloc) 4.96 KB

```
1 import os,random,sys,time
2 from selenium.webdriver.common.keys import Keys
3 from selenium.webdriver.support.ui import WebDriverWait
4 from selenium.webdriver.support import expected_conditions as EC
5 from selenium.webdriver.common.by import By
6 from selenium import webdriver
7 from bs4 import BeautifulSoup
8 from IPython.display import Markdown, display
9
10 def printmd(string):
11     display(Markdown(string))
12
13 driver = webdriver.Chrome('driver/chromedriver.exe')
14
15 driver.get('https://www.linkedin.com/uas/login')
```

```

5
6
7 time.sleep(3)
8 elementID = driver.find_element_by_id('username')
9 print("Email or Phone -")
0 username = input()
1 elementID.send_keys(username)
2
3 elementID = driver.find_element_by_id('password')
4 print("password -")
5 password = input()
6 elementID.send_keys(password)
7
8 elementID.submit()
9
0 print("LinkedIn profile link -")
1 lnk = input()
2 driver.get(lnk)
3
4 print("\n")
5
6 SCROLL_PAUSE_TIME = 2
7 # Get scroll height
8 last_height = driver.execute_script("return document.body.scrollHeight")
9
0 while True:
1     # Scroll down to bottom
2     driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
3
4     # Wait to load page
5     time.sleep(SCROLL_PAUSE_TIME)
6
7     # Calculate new scroll height and compare with last scroll height
8     new_height = driver.execute_script("return document.body.scrollHeight")
9     if new_height == last_height:
0         break
1     last_height = new_height
2
3 time.sleep(5)
4
5 NAME = driver.find_element_by_xpath("//li[@class='inline t-24 t-black t-normal break-words']")
6 print("NAME:- " + NAME.text)
7
8 # CONNECTION_TYPE = driver.find_element_by_xpath("//span[@class='dist-value']")
9 # print("CONNECTION_TYPE:- " + CONNECTION_TYPE.text+" degree connection")
0
1 POSITION = driver.find_element_by_xpath("//h2[@class='mt1 t-18 t-black t-normal break-words']")
2 print("POSITION:- " + POSITION.text)
3
4 LOCATION= driver.find_element_by_xpath("//li[@class='t-16 t-black t-normal inline-block']")

```

```

4 print("LOCATION:- "+ LOCATION.text)
5
6
7 time.sleep(3)
8 driver.execute_script("window.scrollTo(0, document.body.scrollHeight/2);")
9 time.sleep(5)
10 WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.CLASS_NAME, "pv-profile-section
11
12 activity = driver.find_element_by_xpath("//a[@data-control-name='recent_activity_details_all']")
13 driver.get(activity.get_attribute('href'))
14
15 WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.CLASS_NAME, "pv-recent-activity
16 nav = driver.find_element_by_xpath("//nav[@class='pv-recent-activity-detail__pills mt4']")
17 articles = nav.find_elements_by_tag_name('button')
18 articles[1].click()
19 time.sleep(3)
20
21
22 url = driver.current_url
23 driver.get(url)
24
25 SCROLL_PAUSE_TIME = 2
26 # Get scroll height
27 last_height = driver.execute_script("return document.body.scrollHeight")
28 while True:
29     # Scroll down to bottom
30     driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
31
32     # Wait to load page
33     time.sleep(SCROLL_PAUSE_TIME)
34
35     # Calculate new scroll height and compare with last scroll height
36     new_height = driver.execute_script("return document.body.scrollHeight")
37     if new_height == last_height:
38         break
39     last_height = new_height
40
41 try:
42     box = []
43     soup = BeautifulSoup(driver.page_source, 'html.parser')
44     divTag = soup.find("div", {"class": "ember-view"})
45     tags = soup.find_all("article", {"class": "pv-post-entity--detail-page-format artdeco-contain
46     for tag in tags:
47         box.append(tag.find('a').get('href'))
48     main_link = "https://www.linkedin.com"
49     for i in box:
50         driver.get(main_link+i)
51
52     SCROLL_PAUSE_TIME = 2
53     # Get scroll height
54     last_height = driver.execute_script("return document.body.scrollHeight")

```

```

1 while True:
2     # Scroll down to bottom
3     driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
4
5     # Wait to load page
6     time.sleep(SCROLL_PAUSE_TIME)
7
8     # Calculate new scroll height and compare with last scroll height
9     new_height = driver.execute_script("return document.body.scrollHeight")
10    if new_height == last_height:
11        break
12    last_height = new_height
13
14    WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.CLASS_NAME, "reader-ar
15    a = driver.find_element_by_xpath("//h1[@dir='ltr']")
16    print("Heading:-"+" "+a.text)
17    print("\n")
18
19    WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.CLASS_NAME, "reader-ar
20    a = driver.find_element_by_xpath("//div[@class='reader-article-content']")
21    paragraphs = a.find_elements_by_tag_name('p')
22
23    for paragraph in paragraphs:
24        print(paragraph.text)
25
26    print("***END**")
27    print("\n")
28
29 except:
30    print("***WARNING:-you have close the chrome test browser or no article post yet.**")

```