Попытка номер пять по парсеру Facebook.

Шаг №1

Если не использовать аккаунты, через некоторое время могут забанить по IP. Смена IP не всегда помогает, также возможна блокировка по User-Agent или по самому серверу. (Эти созданные предложение пока не понятные)

Если у вас не получается зарегистрировать аккаунт, используйте этот существующий аккаунт - 79896318901:mama230260 (login:password)(Залогинился по предоставленному аккаунту)

Шаг №2

Чтобы авторизоваться использовала код, который был создан в один из уроков.  
from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

from selenium.webdriver.chrome.options import Options

from gologin import GoLogin

from bs4 import BeautifulSoup

import time

token = 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiI2NWNhODdiZmE0YzM0YTAzOGRjYmZkMDkiLCJ0eXBlIjoiZGV2Iiwiand0aWQiOiI2NWNhODhiODAyZDlmZjFkMTI1ODliYTAifQ.ry3kq0hFKnCQmYh93Km9EgJwXYGOtWwu5xQExm20s-Q'

profile\_id = '65ca8a05e0878026237463c9'

port = 3942

gl = GoLogin({

'token': token,

'profile\_id': profile\_id,

'port': port

})

login\_1 = '79896318901'

pass\_1 = 'mama230260 '

chrome\_driver\_path = 'chromedriver.exe'

debugger\_address = gl.start()

chrome\_options = Options()

chrome\_options.add\_experimental\_option('debuggerAddress', debugger\_address)

driver = webdriver.Chrome()

driver.get('http://www.facebook.com')

data\_one = driver.page\_source

# по моей логике – это означает правильно ли я мыслю в эту сторону

# login\_to\_facebook – название функции, driver, login\_1, pass\_1 – принимающие переменные

def login\_to\_facebook(driver, login\_1, pass\_1):

# login\_url – переменная который содержит страницу

login\_url = 'https://www.facebook.com/login.php'

# по моей логике driver.get – возвращает данные html страницы

driver.get(login\_url)

# это ошибки показывает

try:

# email\_element – переменная много чего выполняет но непонятно

# EC.presence\_of\_element\_located((By.ID, ‘email’)) – это условие, которое говорит Selenium ждать, пока на странице не будет найден элемент с указанным id ‘email’.

email\_element = WebDriverWait(driver, 10).until(

EC.presence\_of\_element\_located((By.ID, 'email'))

)

# send\_keys() – метод в SeleniumWebDriver используется для ввода текста или нажатия клавиш в текстовое поле или другой элемент веб-страницы.

email\_element.send\_keys(login\_1)

# password\_element – переменная которая ищет элемента с id ‘pass’

password\_element = driver.find\_element(By.ID, 'pass')

# send\_keys – это метод в SeleniumWebDriver используется для ввода текста или нажатия клавиш в текстовое поле или другой элемент веб-страницы.

password\_element.send\_keys(pass\_1)

# login\_button – переменная которая ищет элемента с id ‘loginbutton’

login\_button = driver.find\_element(By.ID, 'loginbutton')

# click – нажатие кнопки

login\_button.click()

# время на 3 секунды останавливает выполнение авторизации

time.sleep(3)

print('Авторизация прошла успешно!!!')

# при выполнении кода в теле try возникла ошибка в теле except сообщает об ошибке

except Exception as e:

print('Ошибка при авторизации:', e)

# parse\_data – название функции, которая принимает два переменных driver, data\_one – но пока непонятно #откуда идет эти переменные

def parse\_data(driver, data\_one):

if 'Facebook помогает вам всегда оставаться на связи' in str(data\_one) or 'Забыли пароль' in str(data\_one):

# вызывается функция ранее обявленный под названием login\_to\_facebook

login\_to\_facebook(driver, login\_1, pass\_1)

else:

print('Уже авторизован!!!')

# link – это для содержание ссылки которая в своем теле содержит переменную tag\_name передает значение #искомого значения может быть могу ошибаться filters это напонятно

link = f'https://www.facebook.com/groups/gorodpavlodarkz/posts/8066253500075072/'

try:

# data\_ - переменная который содержит у себя функцию get\_data – данная функция где объявлено #непонятно где вы при этом он принимает два аргумента link, driver

data\_ = get\_data(link, driver)

# items - переменная который содержит у себя функцию get\_links\_from\_source – данная функция где #объявлено непонятно где вы при этом он принимает один аргумент data\_ - который объявлено строка выше #принимающая функцию – этот момент надо разобраться

items = get\_title\_from\_source(data\_)

# items – нам возвращает полученные ссылки и в цикле возвращая переменной url пошагово

for title in items:

# post\_url – переменная также использует функции в нашем случае process\_url – принимает себе #переменную в итерации url

post\_title = process\_title(title)

print('Post title:', post\_title)

# except – тело функции который в случае ошибки в теле try выполняет показывает ошибку

except Exception as ex:

print('Main', ex)

# driver – закрывает переменную открытую браузер

driver.close()

# вот и наш долгожданная функция get\_data, который мы искали, оказывается он у нас принимает два #аргумента url, driver

def get\_data(url, driver):

driver.get(url)

for i in range(1, 5):

driver.execute\_script('window.scrollTo(0, document.body.scrollHeight);')

time.sleep(2)

data = driver.page\_source

data = str(data)

return data

def get\_title\_from\_source(data):

soup = BeautifulSoup(data, 'html.parser')

content\_text = soup.find('div', class\_='xdj266r x11i5rnm xat24cr x1mh8g0r x1vvkbs x126k92a').string

print(content\_text)

date\_post = soup.find('span',

class\_='html-span xdj266r x11i5rnm xat24cr x1mh8g0r xexx8yu x4uap5 x18d9i69 xkhd6sd x1hl2dhg x16tdsg8 x1vvkbs').string

print(date\_post)

link = soup.find('div',

class\_='html-div xe8uvvx xdj266r x11i5rnm xat24cr x1mh8g0r xexx8yu x4uap5 x18d9i69 xkhd6sd x1n2onr6').string

print(link)

metrics = soup.find('div',

class\_='x1i10hfl x1qjc9v5 xjbqb8w xjqpnuy xa49m3k xqeqjp1 x2hbi6w x13fuv20 xu3j5b3 x1q0q8m5 x26u7qi x972fbf xcfux6l x1qhh985 xm0m39n x9f619 x1ypdohk xdl72j9 x2lah0s xe8uvvx xdj266r x11i5rnm xat24cr x1mh8g0r x2lwn1j xeuugli xexx8yu x4uap5 x18d9i69 xkhd6sd x1n2onr6 x16tdsg8 x1hl2dhg x1ja2u2z x1t137rt x1o1ewxj x3x9cwd x1e5q0jg x13rtm0m x3nfvp2 x1q0g3np x87ps6o x1lku1pv x1a2a7pz x5ve5x3').string

print(metrics)

filename = 'result.json'

with open(filename, "w", encoding='UTF-8') as file:

file.write("\nТекст поста:\n" + str(content\_text))

file.write("\nДата поста в Facebook:\n" + str(date\_post))

file.write("\nСсылка на вложение:\n" + str(link))

file.write("\nМетрика:\n" + str(metrics))

def process\_title(title):

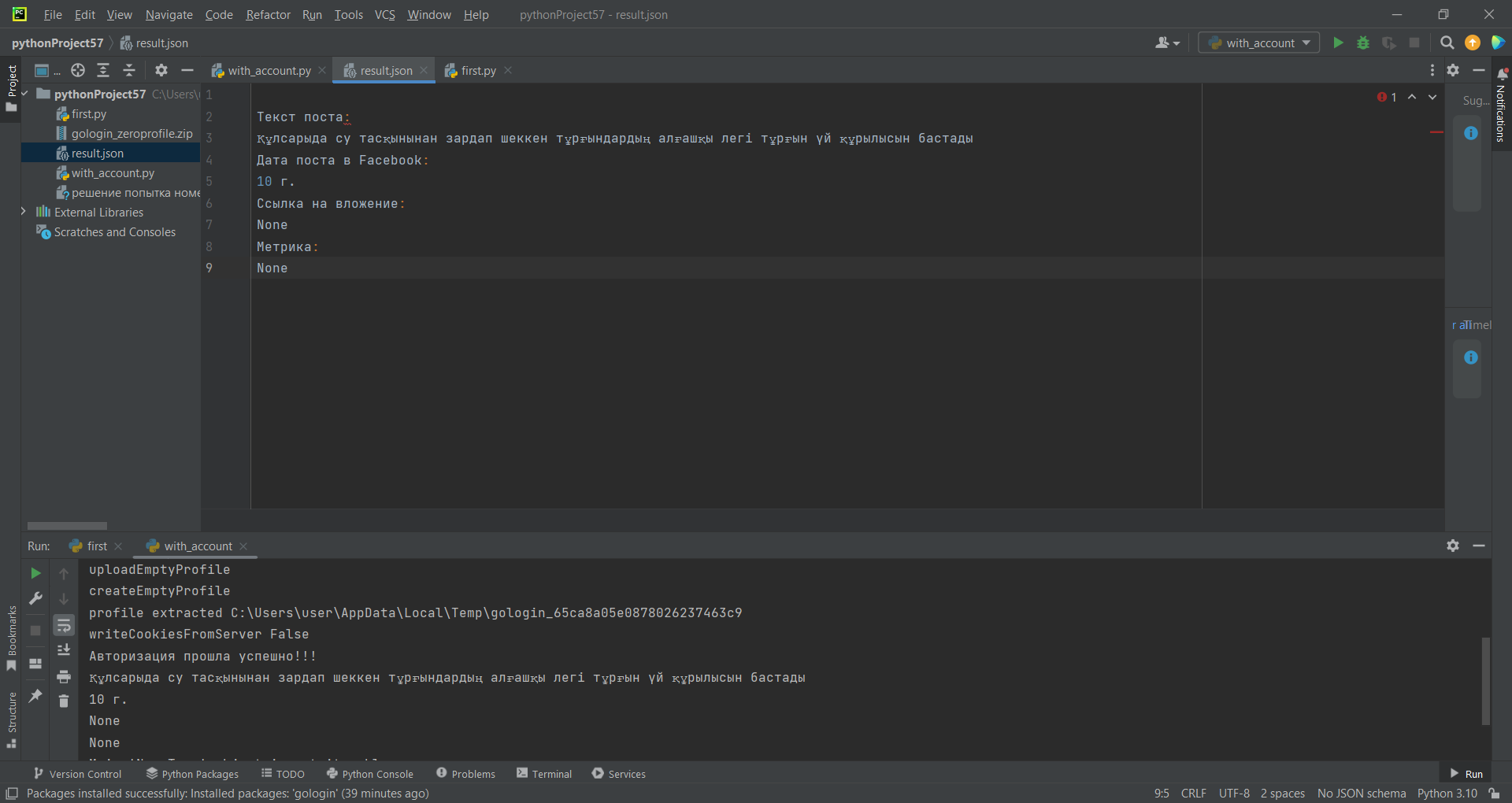
return title

if \_\_name\_\_ == '\_\_main\_\_':

parse\_data(driver, data\_one)

Шаг №3

Но ответы не совсем то что надо. По моей логике ищу определенный тег по классу и оттуда вытаскиваю нужные данные. Везде использую данную логику, но ответы дается где то правильные где то неправильные (Что я делаю не так)



***Например:***

content\_text = soup.find('div', class\_='xdj266r x11i5rnm xat24cr x1mh8g0r x1vvkbs x126k92a').string

print(content\_text) (Эти строки кода дает верные данные)

По моей логике следующие строки кода должны дать правильные данные, но нет. Ответ 10 г. и None

date\_post = soup.find('span',

class\_='html-span xdj266r x11i5rnm xat24cr x1mh8g0r xexx8yu x4uap5 x18d9i69 xkhd6sd x1hl2dhg x16tdsg8 x1vvkbs').string

print(date\_post)

link = soup.find('div',

class\_='html-div xe8uvvx xdj266r x11i5rnm xat24cr x1mh8g0r xexx8yu x4uap5 x18d9i69 xkhd6sd x1n2onr6').string

print(link)

metrics = soup.find('div',

class\_='x1i10hfl x1qjc9v5 xjbqb8w xjqpnuy xa49m3k xqeqjp1 x2hbi6w x13fuv20 xu3j5b3 x1q0q8m5 x26u7qi x972fbf xcfux6l x1qhh985 xm0m39n x9f619 x1ypdohk xdl72j9 x2lah0s xe8uvvx xdj266r x11i5rnm xat24cr x1mh8g0r x2lwn1j xeuugli xexx8yu x4uap5 x18d9i69 xkhd6sd x1n2onr6 x16tdsg8 x1hl2dhg x1ja2u2z x1t137rt x1o1ewxj x3x9cwd x1e5q0jg x13rtm0m x3nfvp2 x1q0g3np x87ps6o x1lku1pv x1a2a7pz x5ve5x3').string

print(metrics) (Что я делаю не так)