Міністерство освіти і науки України

Національний технічний університет України

"Київський політехнічний інститут імені Ігоря Сікорського"

Фізико-технічний інститут

WEB-АНАЛІТИКА

Лабораторна робота

Робота №3

Виконав студент гр. ФБ-31мп:

Шевченко С. Ш.

Київ – 2024

[Репозиторій](https://github.com/dedkuzmich/web-analytics-2024) Github з кодом.

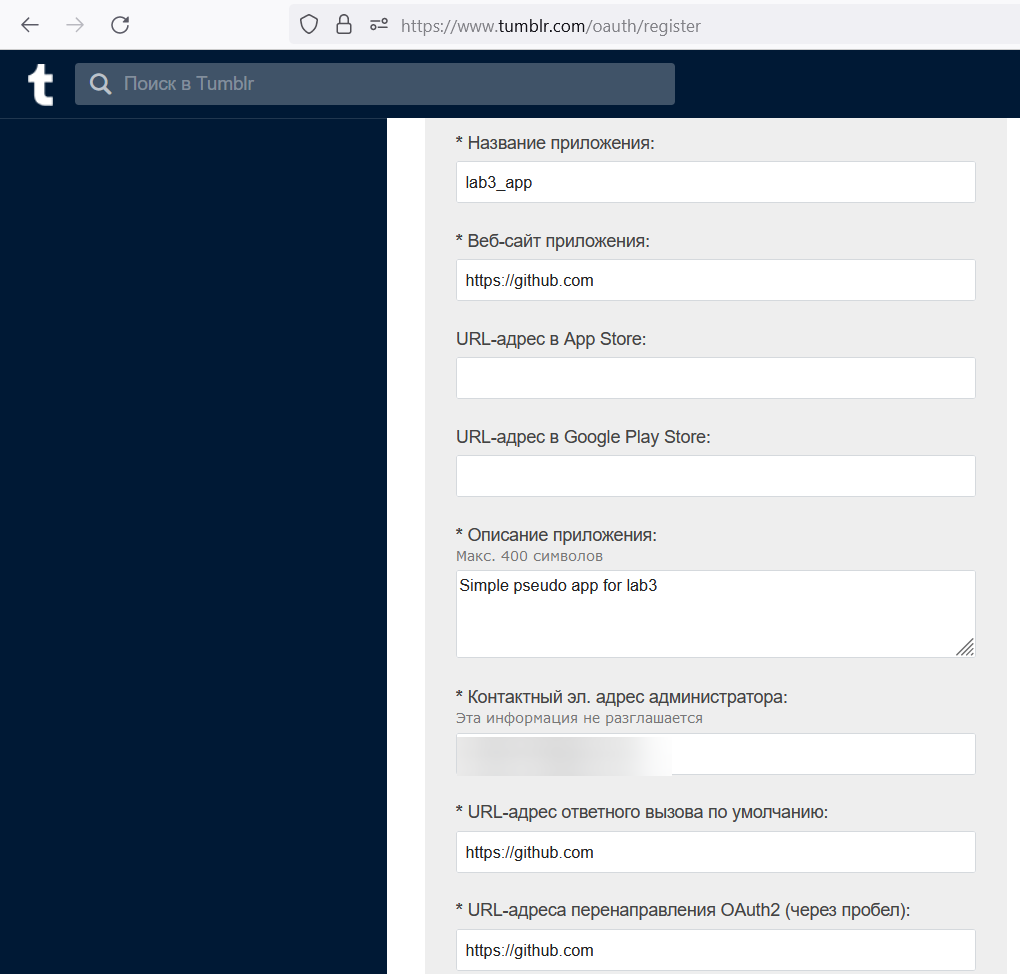
# Завдання:

Використовуючи API Twitter, написати скрипт, що буде автентифікуватися в соціальній мережі та забирати N>1000 останніх записів певного акаунту (акаунт має бути публічним), та створити датасет. Опціонально – сформувати систему сповіщення (пошта, телефон, будь-що) про знаходження певного ключового слова.

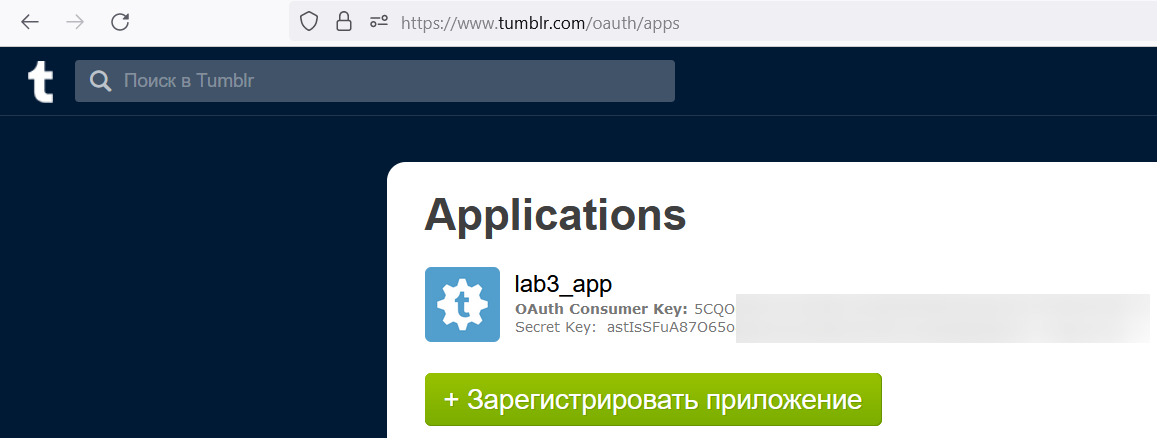
# Виконання завдань:

Соц. мережа X (колишній Twitter) обмежує необхідні запити до API різних версій. В якості цільової соц. мережі було обрано Tumblr.

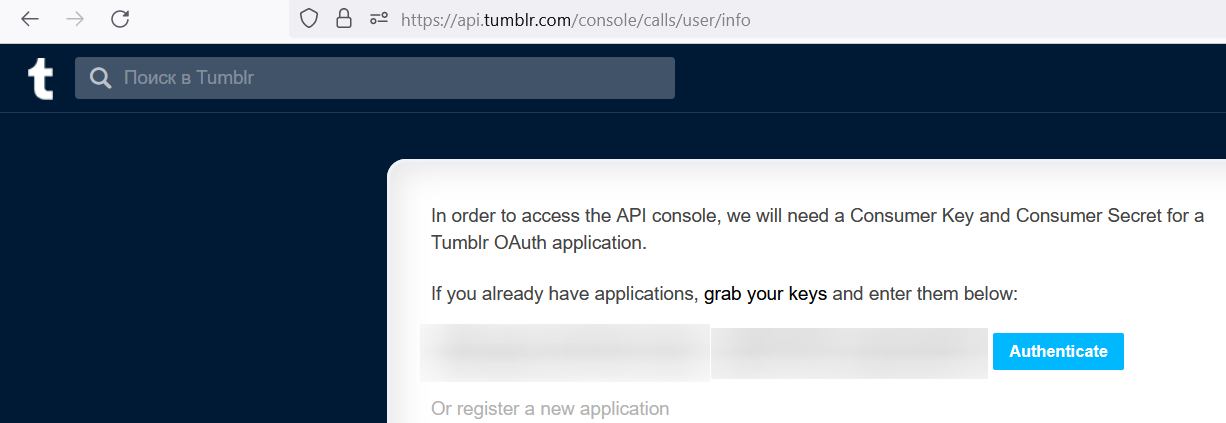
[Реєструємо](https://www.tumblr.com/oauth/register) додаток lab3\_app:

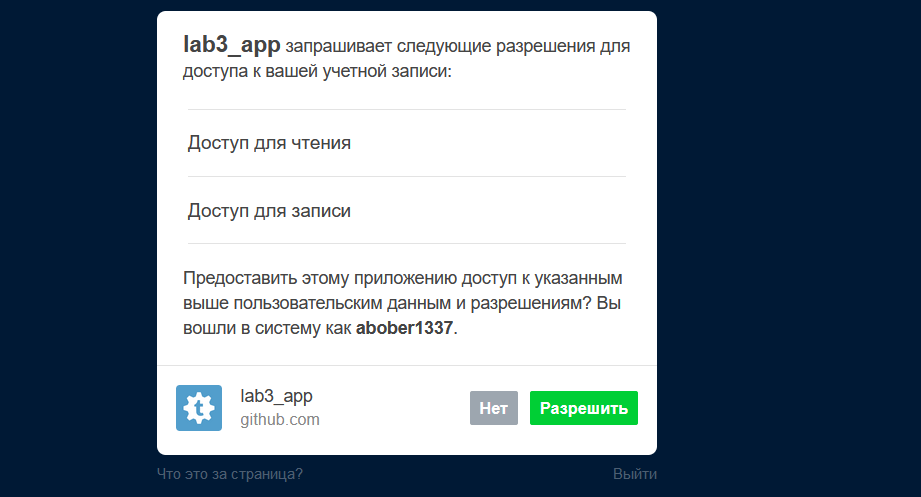


Після реєстрації [бачимо](https://www.tumblr.com/oauth/apps) новостворений додаток та 2 ключі (OAuth Consumer Key та Secret Key):

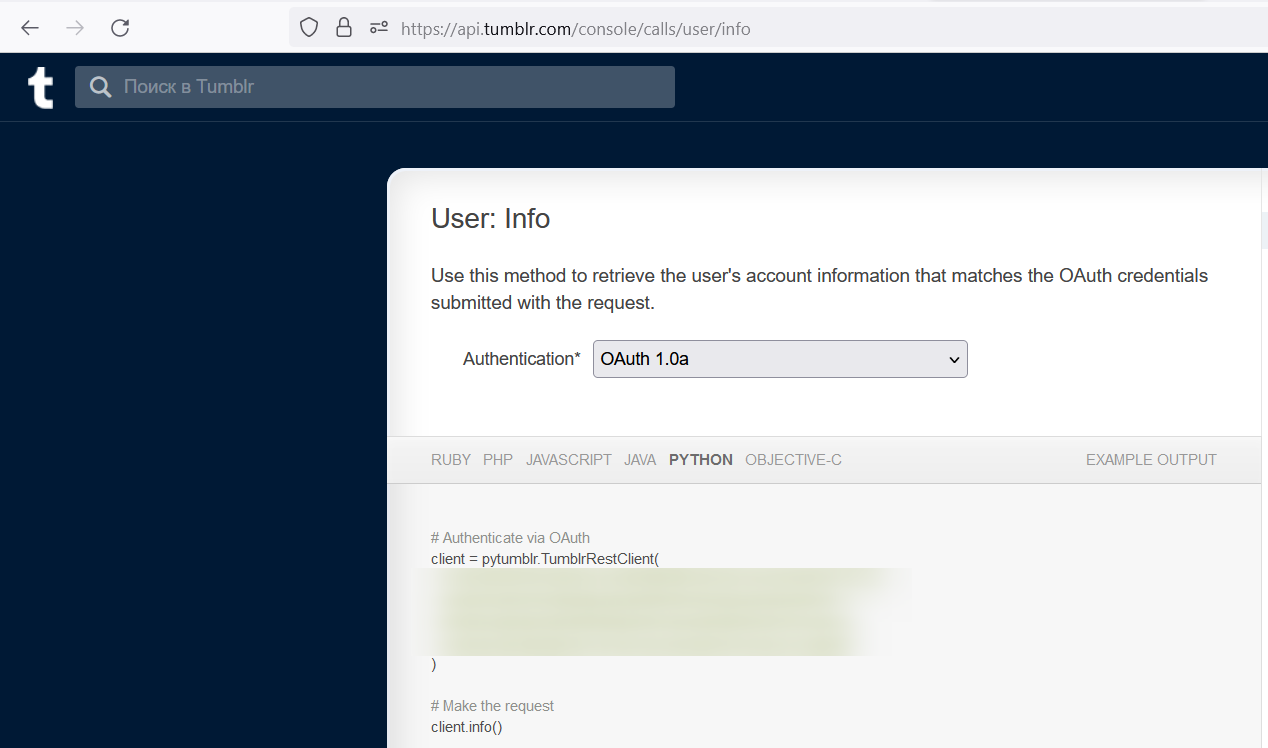


[Отримаємо](https://api.tumblr.com/console/calls/user/info) дані для Oauth:





Бачимо приклад використання клієнта OAuth в Python:



В якості цільового акаунту було обрано [redglassbird](https://www.tumblr.com/redglassbird).

Автентифікуємося в Tumblr через API (як у прикладі вище). Дані для входу беруться з файлу credentials.json, вміст якого буде приведено неповністю з міркувань безпеки:

# Authenticate via Tumblr API  
*with* open("credentials.json") *as* fin:  
 credentials = json.load(fin)  
  
CONSUMER\_KEY = credentials["CONSUMER\_KEY"]  
CONSUMER\_SECRET = credentials["CONSUMER\_SECRET"]  
OAUTH\_TOKEN = credentials["OAUTH\_TOKEN"]  
OAUTH\_SECRET = credentials["OAUTH\_SECRET"]  
  
client = pytumblr.TumblrRestClient(CONSUMER\_KEY, CONSUMER\_SECRET, OAUTH\_TOKEN, OAUTH\_SECRET)

Тепер треба зчитати 1050 (>1000) дописів користувача redglassbird. За один запит до API можемо зчитати тільки 50 дописів, тобто знадобиться 1050 / 50 = 21 запит.

Кожен допис містить дуже багато полів, але ми відбираємо тільки 5:

1) Назву блогу

2) Дату та час у відповідному форматі

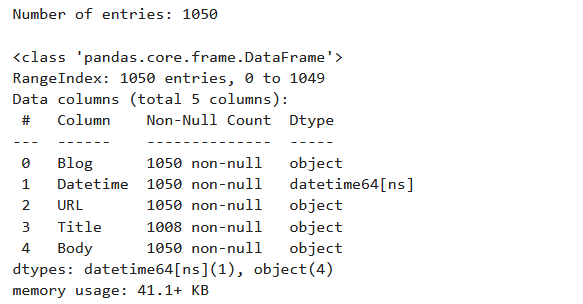
3) Посилання на допис

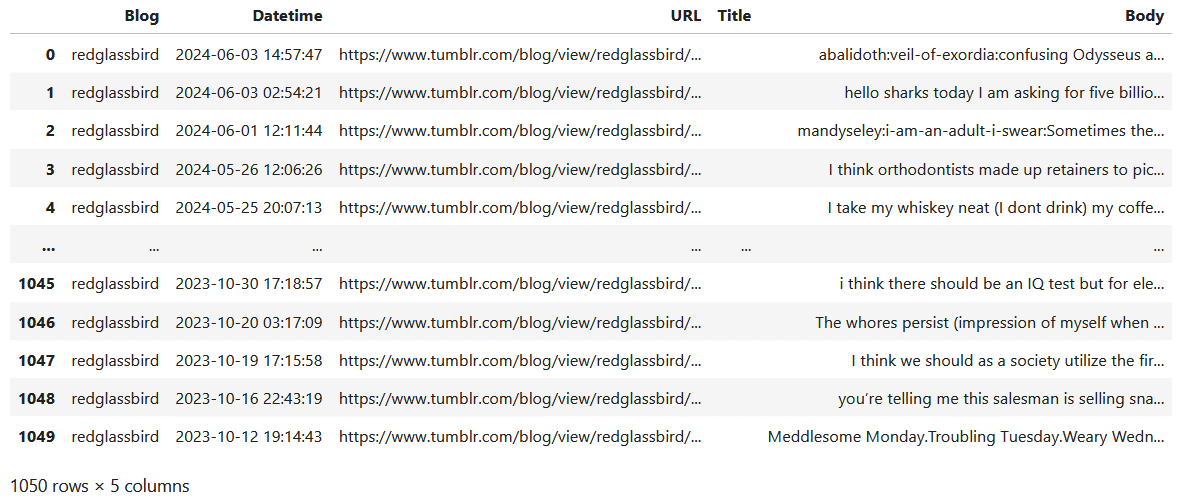
4) Назву допису

5) Вміст допису, який поданий у виді HTML і який треба розпарсити.

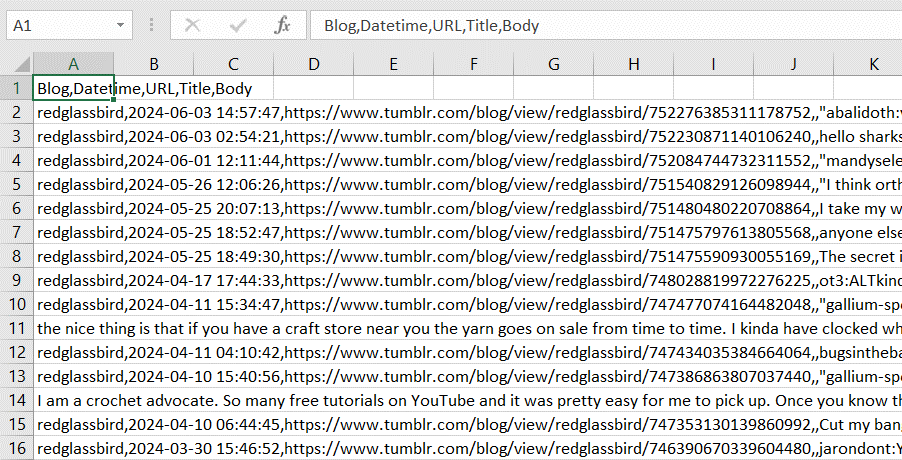
# Retrieve posts from Tumblr and create dataset  
blog\_name = "redglassbird" # Source: https://www.tumblr.com/redglassbird  
num\_posts = 1050 # Total number of posts  
limit = 50 # Max number of posts that can be retrieved in a single request  
list\_posts = []  
*for* i *in* range(num\_posts // limit):  
 posts = client.posts(blog\_name, limit = limit)  
 *for* post *in* posts["posts"]:  
 # Parse Datetime  
 datetime\_str = post["date"]  
 datetime = dt.strptime(datetime\_str, "%Y-%m-%d %H:%M:%S %Z")  
  
 # Parse HTML body as text  
 html\_body = post["body"]  
 soup = BeautifulSoup(html\_body, "html.parser")  
 body = soup.get\_text()  
  
 dict\_post = {  
 "Blog": post["blog\_name"],  
 "Datetime": datetime,  
 "URL": post["post\_url"],  
 "Title": post["title"],  
 "Body": body  
 }  
 list\_posts.append(dict\_post)  
  
# Save dataset as CSV  
file\_csv = f"tumblr-{blog\_name}.csv"  
df = pd.DataFrame(list\_posts)  
df.to\_csv(file\_csv, index = *False*)  
print(f"Number of entries: {len(df)}\n")  
df.info()  
display(df)

Отримаємо датасет на 1050 рядків:





Датасет збережено у файл tumblr-redglassbird.csv:



...

