



Social Media Data Streaming

---- Kayla Liu



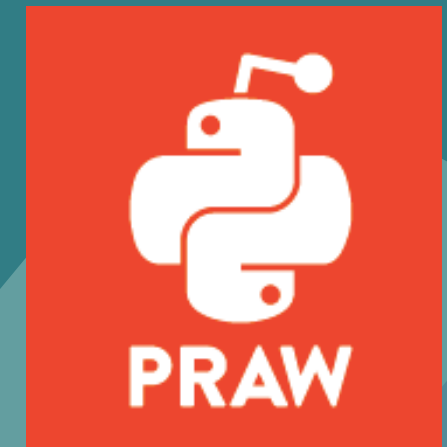
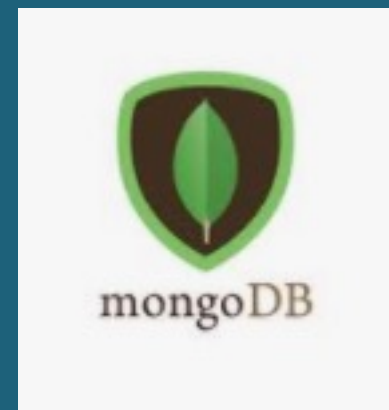
Contents

1. Data Collection
 2. Data Model
 3. Data Storage
 4. Build Web Application
- 

Social Media



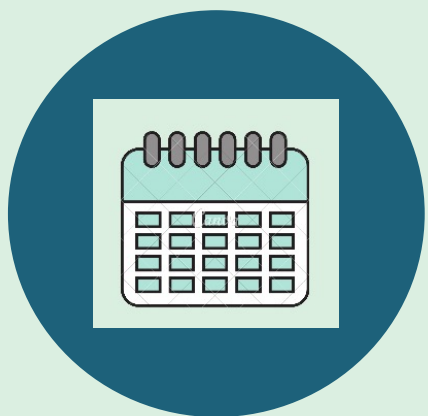
Tools



Data Model



```
post =[ ]
for d in tweets_text:
    postTwitter = {
        'id': user.id,
        'name':user.name ,
        'text': d
    }
```



```
for posts in postReddit:
    posts={ "id":posts.id,
            "name": posts.author.name,
            "Text": posts.title}
```


➤ Data Collection - Twitter

```
hellotwitter.py x helloreddit.py
venv > hellotwitter.py > ...
1 import tweepy
2 import dns
3 from dns import resolver
4 from pymongo import MongoClient
5
6
7 client = MongoClient('mongodb+srv://Yuxin:Lyx19960227@cluster0.nhbu1.mongodb.net/myFirstDatabase?retryWrites=true&ssl=true')
8 db = client['Assignment2']
9
10 # Variables that contains the user credentials to access Twitter API
11 ACCESS_TOKEN = "1406070447481884672-Kqtr6XxTAq3lcghh0jFUrHDCGUknNj"
12 ACCESS_TOKEN_SECRET = "CYwxFMEqTiLDvlHIA2ExcUDwN9IxlclFqISSFsgZyAJC0"
13 CONSUMER_KEY = "WS8tQxD1P1GgmSgTWCH8J4Ma0"
14 CONSUMER_SECRET = "QIU8lBApulLocyIgCXJJSEzXVUiev63CgnE3Qom6UiYnn53All"
15
16 def stream_tweets(username):
17     # This handles Twitter authentication and the connection to Twitter Streaming API
18     auth= tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
19     auth.set_access_token(ACCESS_TOKEN,ACCESS_TOKEN_SECRET)
20
21
22     # Create an API object
23     api = tweepy.API(auth)
24
25     # Read the tweets
26     tweets = api.user_timeline(screen_name=username)
27     print_tweets(username)
```

```
hellotwitter.py x helloreddit.py
venv > hellotwitter.py > ...
18     auth= tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
19     auth.set_access_token(ACCESS_TOKEN,ACCESS_TOKEN_SECRET)
20
21
22     # Create an API object
23     api = tweepy.API(auth)
24
25     # Read the tweets
26     tweets = api.user_timeline(screen_name=username)
27     user = api.get_user(username)
28     tweets_text = [tweet.text for tweet in tweets]
29
30     # Store in the database
31     post =[]
32     for d in tweets_text:
33         postTwitter = {
34             'id': user.id,
35             'name':user.name ,
36             'text': d
37         }
38         post.append(postTwitter)
39
40     print(postTwitter)
41     twitter1 = db.Twitter
42     sub_id = twitter1.insert_many(post)
43
44 if __name__ == '__main__':
45     stream_tweets("YuxinLi19980652")
46
```

REFERENCE: <https://www.tweepy.org/>

➤ Data Collection - Reddit

```
hellotwitter.py  helloreddit.py x
venv > helloreddit.py > ...
1  import praw
2  import requests
3  import dns
4  from dns import resolver
5  from pymongo import MongoClient
6
7  client = MongoClient('mongodb+srv://Yuxin:Lyx19960227@cluster0.nhbu1.mongodb.net/myFirstDatabase?retryWrites=true&w=majority')
8
9  # Variables that contains the user credentials to access reddit API
10 reddit = praw.Reddit(client_id = "n6gNZGiFkrNPqw",
11                      client_secret = "--Dt4efhzMRDqg1CPtumJ17TLX3FUA",
12                      user_agent = "YuxinLiu123",
13                      username = "YuxinLiu123",
14                      password = "Lyx19960227")
15
16 post = [ ]
17
18 postReddit = reddit.redditor("YuxinLiu123").submissions.top()
19 for posts in postReddit:
20     posts={ "id":posts.id,
21            "name": posts.author.name,
22            "Text": posts.title}
23
24     post.append(posts)
25
26
27 db = client['Assignment2']
28 subreddit1 = db.Reddit
29
30 subreddit1.insert_many(post)
```

REFERENCE: <https://praw.readthedocs.io/en/latest/>

➤ MongoDB- Twitter

Twitter

FILTER {"filter":"example"}

QUERY RESULTS 1-5 OF 5

_id: ObjectId("60d073319d385c1e8d1a2768")
id: 1406070447481884672
name: "Yuxin Liu"
text: "123"

_id: ObjectId("60d073319d385c1e8d1a2769")
id: 1406070447481884672
name: "Yuxin Liu"
text: "Post to Twitter"

_id: ObjectId("60d073319d385c1e8d1a276a")
id: 1406070447481884672
name: "Yuxin Liu"
text: "interesting news"

_id: ObjectId("60d073319d385c1e8d1a276b")
id: 1406070447481884672
name: "Yuxin Liu"
text: "RT @NASA: 🚀 Want more space in your inbox? Sign up for the NASA Exp..."

➤ MongoDB- Reddit

The screenshot displays the MongoDB Compass web interface. On the left sidebar, under the 'Namespaces' section, the 'Assignment2' database is expanded, showing the 'Reddit' collection selected. The main panel is titled 'Assignment2.Reddit' and shows collection statistics: 'COLLECTION SIZE: 434B', 'TOTAL DOCUMENTS: 4', and 'INDEXES TOTAL SIZE: 4KB'. Below the statistics, there are tabs for 'Find', 'Indexes', 'Schema Anti-Patterns', 'Aggregation', and 'Search Indexes'. The 'Find' tab is active, showing a filter bar with the text '{\"filter\":\"example\"}'. Below the filter, it says 'QUERY RESULTS 1-4 OF 4'. The results are displayed in a list of documents. The first document is expanded, showing its fields: '_id: ObjectId(\"60d073834b5de9cfddb0ebf6\")', 'id: \"o45glb\"', 'name: \"YuxinLiu123\"', and 'Text: \"have a nice day\"'. The second document is partially visible, showing '_id: ObjectId(\"60d073834b5de9cfddb0ebf7\")', 'id: \"o2p7t0\"', 'name: \"YuxinLiu123\"', and 'Text: \"No number before 1,000 contains the letter A\"'. The third document is also partially visible, showing '_id: ObjectId(\"60d073834b5de9cfddb0ebf8\")'.

+ Create Database

Q NAMESPACES

Assignment2

Reddit

Twitter

Assignment2.Reddit

COLLECTION SIZE: 434B TOTAL DOCUMENTS: 4 INDEXES TOTAL SIZE: 4KB

Find Indexes Schema Anti-Patterns 0 Aggregation Search Indexes ●

FILTER {\"filter\":\"example\"}

QUERY RESULTS 1-4 OF 4

```
{
  "_id": ObjectId("60d073834b5de9cfddb0ebf6"),
  "id": "o45glb",
  "name": "YuxinLiu123",
  "Text": "have a nice day"
}
```

> {
 "_id": ObjectId("60d073834b5de9cfddb0ebf7"),
 "id": "o2p7t0",
 "name": "YuxinLiu123",
 "Text": "No number before 1,000 contains the letter A"
}

```
{
  "_id": ObjectId("60d073834b5de9cfddb0ebf8")
}
```

➤ Web Application

```
@app.route('/')
def Twitter111():
    if request.args:
        tweet = request.args.get('tweet')

        auth= tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
        auth.set_access_token(ACCESS_TOKEN,ACCESS_TOKEN_SECRET)
        api = tweepy.API(auth)

        #update the status
        api.update_status(status = tweet)

        user0 = api.get_user("YuxinLi19980652")
        Twitter111 = {
            'id': user0.status.id,
            'name': user0.name,
            'image_url': user0.profile_image_url,
            'content':tweet
        }
        twitter1 = db.Twitter
        twitter1.insert_one(Twitter111)
        result = True
    else:
        result = False
    return render_template('form.html', res = result )
```

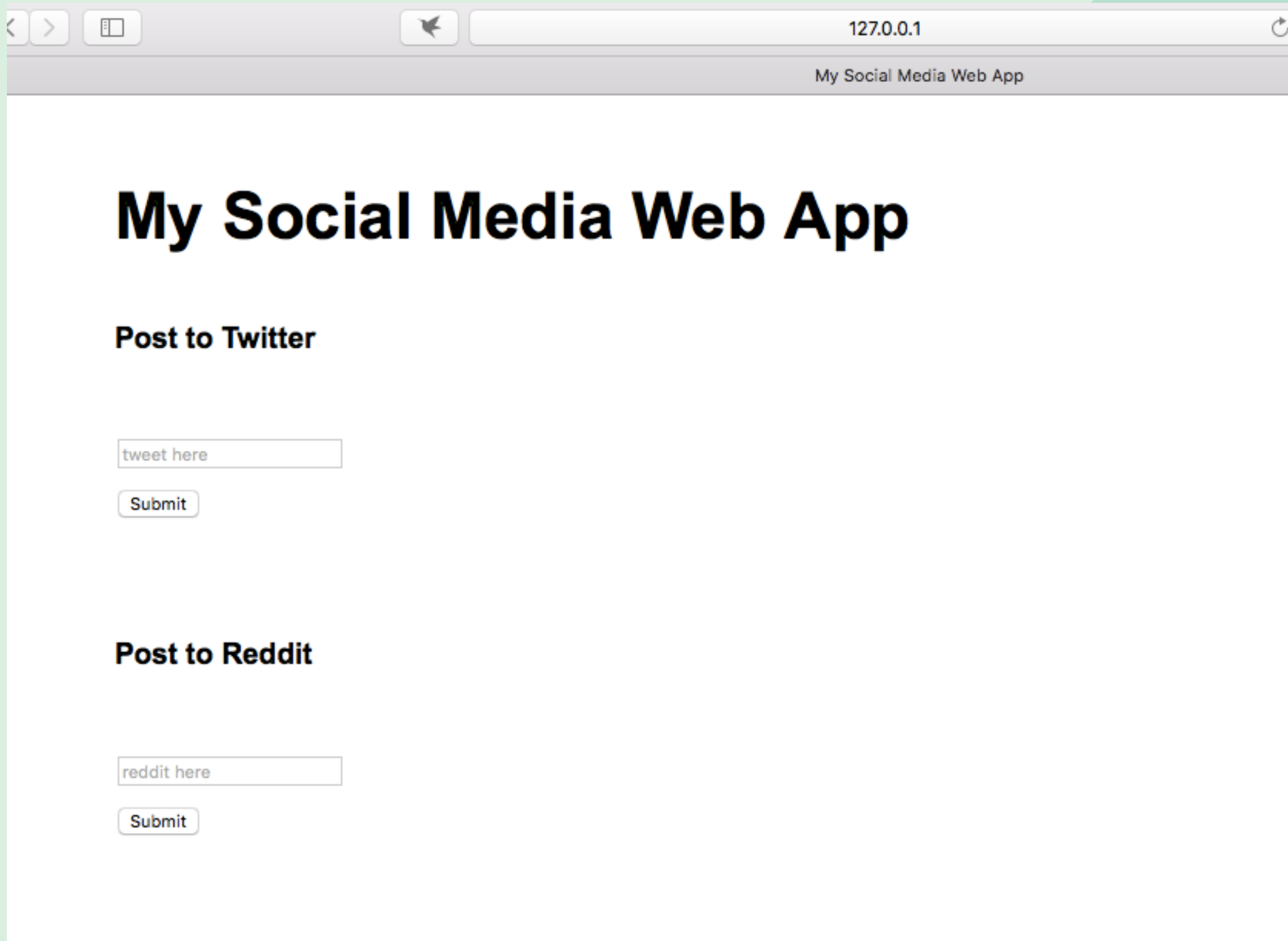
```
@app.route('/reddit')
def Reddit111():
    if request.args:
        post = request.args.get('post')
        sub = reddit.subreddit('u_YuxinLiu123')

        result = True
        subreddit = {
            "id": sub.id,
            "name": "YuxinLiu123",
            "Text":post
        }
        subreddit1= db.Reddit
        subreddit1.insert_one(subreddit)
        result = True
    else:
        result = False
    return render_template('form.html', res = result )

if __name__ == '__main__':
    app.run(debug=True)
```

REFERENCE: <https://www.youtube.com/watch?v=9RAA845Gxn8>

➤ Web page



A screenshot of a web browser window displaying a web application titled "My Social Media Web App". The browser's address bar shows the URL "127.0.0.1". The application interface features a main heading "My Social Media Web App" in a large, bold, black font. Below this heading, there are two distinct sections for posting to social media. The first section, titled "Post to Twitter", contains a text input field with the placeholder text "tweet here" and a "Submit" button. The second section, titled "Post to Reddit", contains a text input field with the placeholder text "reddit here" and another "Submit" button. The entire application is presented on a clean white background within the browser window.

127.0.0.1

My Social Media Web App

My Social Media Web App

Post to Twitter

Post to Reddit

➤ HTML/ CSS

```
nv > templates > form.html > html > body
1
2 <!DOCTYPE html>
3 <html lang="en">
4   <head>
5     <title>My Social Media Web App </title>
6     <link rel="stylesheet" href="/static/style.css" />
7   </head>
8   <body>
9
10    <h2>My Social Media Web App</h2>
11
12    <div class="container m-5">
13      <h6>Post to Twitter</h6>
14      <form type="POST" action="/">
15        <div class="row">
16          <div class="col">
17            <input type="text" class="form-control" placeholder="tweet here" name=
18          </div>
19          <input type="submit" name="Post to Twitter">
20        </div>
21      </form>
22
```

```
nv > static > style.css > body
1 html {
2   font-size: 28px;
3   font-family: "Segoe UI", Arial, Helvetica, sans-serif;
4 }
5
6 body {
7   max-width: 1000px;
8   margin: 0 auto;
9   padding: 20px;
10 }
11
12 .form__fieldset {
13   padding: 20px 10px;
14 }
15
16 .form__fieldset:not(:last-of-type) {
17   border-bottom: 1px solid rgba(0, 0, 0, 0.1);
18 }
19
20 .form__input {
21   font-size: 1rem;
22 }
23
```


Demo

Link to the presentation:

https://www.youtube.com/watch?v=tnehrrtb_Hc



THANKS

—