
Modern NLP

**Based on Deep Learning and Language models.
Day 2 Afternoon**



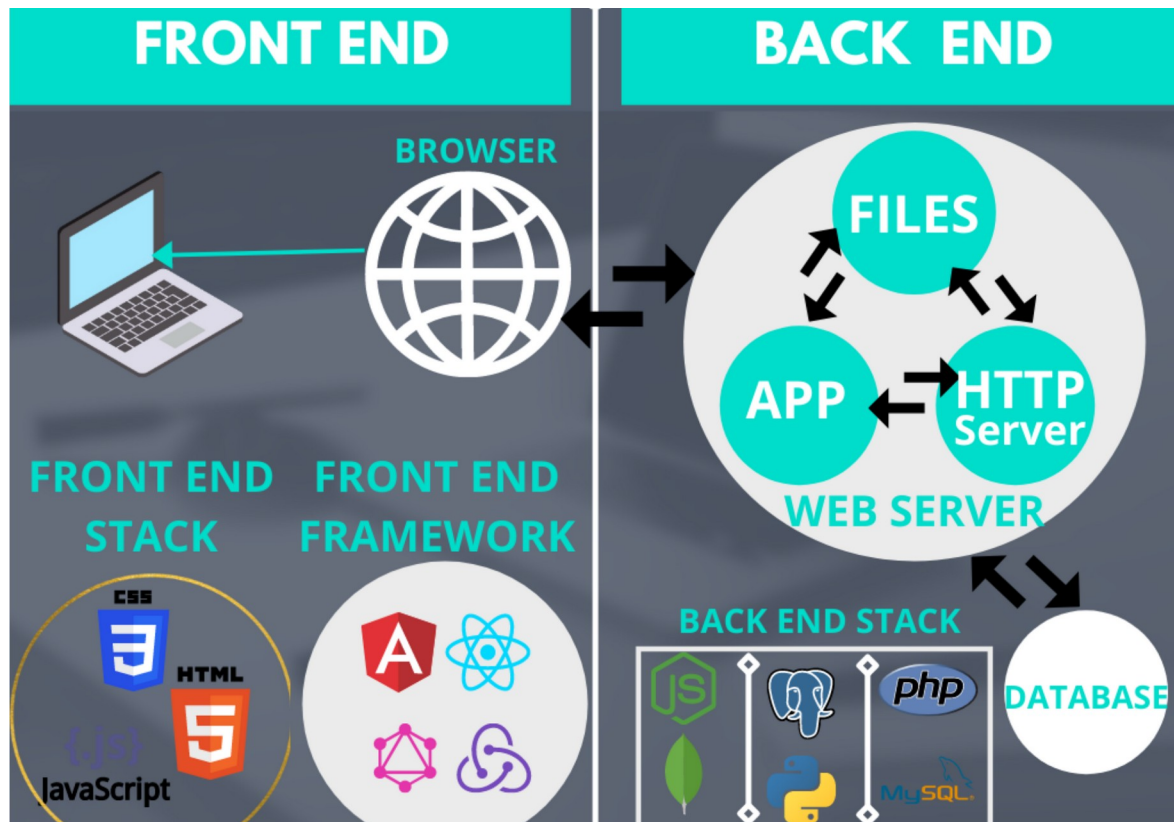
2nd Day

1. Morning (~ 2h)
 - } Smallest remainder of Day 1
 - } King – Man + Woman
 - } Using advanced embedding techniques

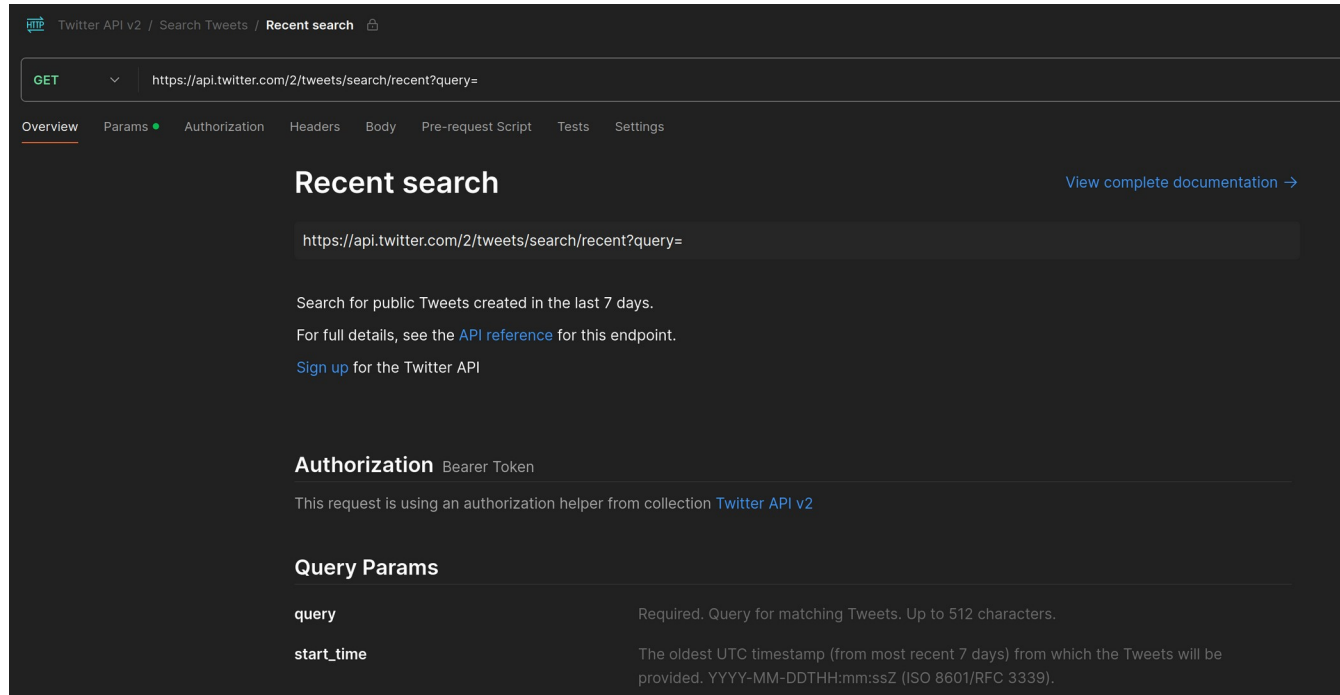
2. Afternoon (~ 4h)
 - } Gentle introduction to delivery API + front End
 - } Transfert learning
 - } Fine Tuning

First ... Let's Talk !

Back vs Front ?



What is Back-End / API ?



The screenshot shows the Swagger UI for the Twitter API v2. The breadcrumb trail is 'Twitter API v2 / Search Tweets / Recent search'. The method is 'GET' and the URL is 'https://api.twitter.com/2/tweets/search/recent?query='. The 'Overview' tab is selected, showing the endpoint name 'Recent search' and a link to 'View complete documentation'. Below this, the URL is repeated in a text box. The description states: 'Search for public Tweets created in the last 7 days. For full details, see the [API reference](#) for this endpoint. [Sign up](#) for the Twitter API'. The 'Authorization' section indicates 'Bearer Token' and mentions an authorization helper from the 'Twitter API v2' collection. The 'Query Params' section lists two parameters: 'query' (Required, Query for matching Tweets. Up to 512 characters.) and 'start_time' (The oldest UTC timestamp (from most recent 7 days) from which the Tweets will be provided. YYYY-MM-DDTHH:mm:ssZ (ISO 8601/RFC 3339)).

Twitter API v2 / Search Tweets / Recent search

GET `https://api.twitter.com/2/tweets/search/recent?query=`

Overview Params Authorization Headers Body Pre-request Script Tests Settings

Recent search [View complete documentation →](#)

`https://api.twitter.com/2/tweets/search/recent?query=`

Search for public Tweets created in the last 7 days.
For full details, see the [API reference](#) for this endpoint.
[Sign up](#) for the Twitter API

Authorization Bearer Token

This request is using an authorization helper from collection [Twitter API v2](#)

Query Params

query	Required. Query for matching Tweets. Up to 512 characters.
start_time	The oldest UTC timestamp (from most recent 7 days) from which the Tweets will be provided. YYYY-MM-DDTHH:mm:ssZ (ISO 8601/RFC 3339).

Fast API

Create it

- Create a file `main.py` with:


```
from typing import Union

from fastapi import FastAPI

app = FastAPI()

@app.get("/")
def read_root():
    return {"Hello": "World"}

@app.get("/items/{item_id}")
def read_item(item_id: int, q: Union[str, None] = None):
    return {"item_id": item_id, "q": q}
```

 Or use `async def ...`



What is Front End

Configuration


Choose your preferred API:


echarts

Choose an example

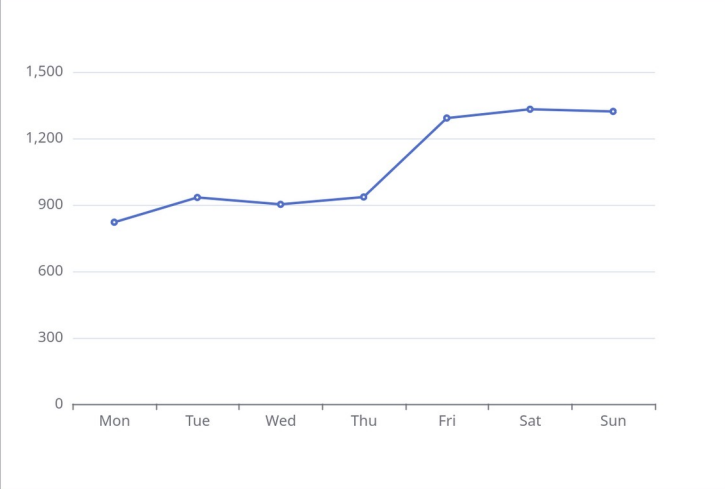
Line: Basic Line Chart

ECharts demos are extracted from <https://echarts.apache.org/examples/en/index.html>, by copying/formatting the 'option' json object into st_echarts. Definitely check the echarts example page, convert the JSON specs to Python Dicts and you should get a nice viz.

Made in  by @andfanilo

 Buy me a coffee

Streamlit ECharts Demo



Day	Value
Mon	800
Tue	950
Wed	900
Thu	950
Fri	1300
Sat	1350
Sun	1350

Source Code

Streamlit

```
import streamlit as st
import pandas as pd
import numpy as np
```

3. Every good app has a title, so let's add one:

```
st.title('Uber pickups in NYC')
```

4. Now it's time to run Streamlit from the command line:

```
streamlit run uber_pickups.py
```


Streamlit

```
import streamlit as st  
import pandas as pd  
import numpy as np
```

3. Every good app has a title, so let's add one:

```
st.title('Uber pickups in NYC')
```

4. Now it's time to run Streamlit from the command line:

```
streamlit run uber_pickups.py
```

Production Back







[Dashboard](#)[Blueprints](#)[Env Groups](#)[Docs](#)[Community](#)[Help](#)[New +](#)

Alexandre GAZAGNES



Overview

[Active 9](#)[Suspended 2](#)[All 11](#)

SERVICE NAME	STATUS	TYPE	RUNTIME	REGION	LAST DEPLOYED ↓	
 mysql8	 Deployed	Web Service	Image	Oregon	5 months ago	...
 mysql	 Failed deploy	Web Service	Image	Oregon	5 months ago	...
 emilie-for-alex	 Failed deploy	Web Service	Python 3	Oregon	5 months ago	...

Production Front

 alexandregazagnes ▾

My apps

Explore

Create app

alexandregazagnes' apps

New app ▾

 audible • main • main.py ★	🌙 🌐 ⋮
 the-jalait-project • dev • main.py ★	🌙 🌐 ⋮
 awdible • bug/14-streamlit-not-working • ./awdible/gui/front.py	🌙 🌐 ⋮
 global-biodiversity-score • main • main.py	🌙 🌐 ⋮

Practice !

(FreeStyle ;))
