Final Project

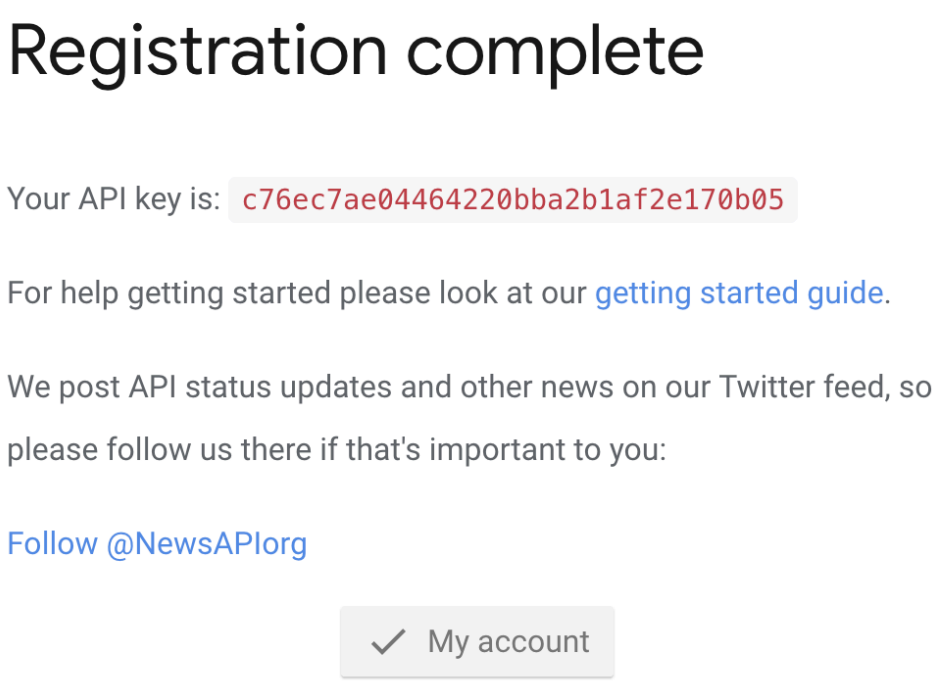
Professor: Saber Amini

Members: Chen, Kun (8977010)

Kalaiarasan, Abinaya (8953115)

Yalamkayala, Haripriya Naga Neelima. (8939330)

Get NewsAPI token



Registration complete

Your API key is: `c76ec7ae04464220bba2b1af2e170b05`

For help getting started please look at our getting started guide.

We post API status updates and other news on our Twitter feed, so please follow us there if that's important to you:

Follow @NewsAPIorg

My account

Start zookeeper & kafka server, then create a new topic: DonaldTrump

```
# in zookeeper/
bin/zkServer.sh start

# in confluent/
nohup bin/kafka-server-start etc/kafka/server.properties > /dev/null 2>&1 &

# Create topic: DonaldTrump
bin/kafka-topics --create --zookeeper localhost:2181 --partitions 1 --replication-factor 1
--topic DonaldTrump
# Show topic describe
bin/kafka-topics --describe --zookeeper localhost:2181 --topic DonaldTrump

```

Setup a Kafka producer

We use the exactly same source code from midterm project.

1. Get all article from NewsAPI which mention **"Donald Trump"**
2. Pass those article to the consumer via kafka,

Producer

```
import requests
from confluent_kafka import Producer
from dotenv import load_dotenv
import os
import time
import hashlib

load_dotenv()
api_key = os.getenv('NEWS_API_KEY')

news_url = f'https://newsapi.org/v2/everything?q=Donald&q=Trump&sortBy=popularity&apiKey=
{api_key}'

producer = Producer({'bootstrap.servers': 'localhost:9092'})
topic = 'DonaldTrump'
processed_ids = set()

def delivery_report(err, msg):
    if err is not None:
        print(f'Message sent failed: {err}')
    else:
        print(f'Message Send: {msg.topic()} [{msg.partition()}]')

while True:
    try:
        response = requests.get(news_url)
        response.raise_for_status()
        news_data = response.json()

        for article in news_data['articles']:
```

```
            url = article.get('url', 'Null')
            title = article.get('title', 'No title')
            publishedAt = article.get('publishedAt', '0')

            source = article['source'].get('name', 'Null')
            author = article.get('author', 'Null')
            description = article.get('description', 'No description')
            urlToImage = article.get('urlToImage', 'Null')
            content = article.get('content', 'Null')

            article_id = hashlib.md5(f"{title}{publishedAt}{url}".encode('utf-8')).hexdigest()
            article['article_id'] = article_id

            message = str(source) +'|'+ str(author) +'|'+ str(description) +'|'+ str(urlToImage) +'|'+ str(content)

            if article_id not in processed_ids:
                processed_ids.add(article_id)
                producer.produce(topic, message.encode('utf-8'), callback=delivery_report)

                producer.flush()
            else:
                continue

        print("Wait Next Pull ... ")
        time.sleep(10)

    except requests.exceptions.RequestException as e:
        print(f'Request Failed: {e}')
        time.sleep(60)
    except Exception as e:
        print(f'Error: {e}')
        break
```

Spark Streaming

In consumer side:

1. Get message from Kafka
2. Process the data
3. Aggregate the data
4. Save the data to HDFS

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import *
from pyspark.sql.types import *

spark = SparkSession.builder.appName("NewsStreamingConsumer") \
    .config("spark.sql.shuffle.partitions", "2") \
    .getOrCreate()
```

```
8
9   spark.sparkContext.setLogLevel("WARN")
10
11  raw_df = spark.readStream.format("kafka") \
12      .option("kafka.bootstrap.servers", "localhost:9092") \
13      .option("subscribe", "DonaldTrump") \
14      .option("startingOffsets", "latest") \
15      .load()
16
17  string_df = raw_df.selectExpr("CAST(value AS STRING) as value")
18
19  split_df = string_df.select(
20      split(col("value"), "\|").getItem(0).alias("source"),
21      split(col("value"), "\|").getItem(1).alias("author"),
22      split(col("value"), "\|").getItem(2).alias("description"),
23      split(col("value"), "\|").getItem(3).alias("urlToImage"),
24      split(col("value"), "\|").getItem(4).alias("content"),
25      current_timestamp().alias("timestamp")
26  )
27
28  aggregated_df = split_df \
29      .withWatermark("timestamp", "1 minute") \
30      .groupBy(window(col("timestamp"), "1 minute"),col("source")).count()
31
32  query = aggregated_df.select(col("window").start.alias("window_start"), \
33          col("window").end.alias("window_end"), col("source"), col("count")) \
34      .writeStream.outputMode("append").format("csv") \
35      .option("path", "/BigData/news_data/") \
36      .option("checkpointLocation", "/BigData/news_checkpoint") \
37      .start()
38
39  query.awaitTermination()
```

Start producer

```
aiml_group77@cluster-6ad7-m:~/midterm-project$ python producer.py
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
Message Send: DonaldTrump [0]
```

Start Spark Streaming

```
gibdata8451@cluster-e861-m:~/final_test$ spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.12:3.4.1 spark_consumer_hdfs.py
:: loading settings :: url = jar:file:/usr/lib/spark/jars/ivy-2.5.1.jar!/org/apache/ivy/core/settings/ivysettings.xml
Ivy Default Cache set to: /home/gibdata8451/.ivy2/cache
The jars for the packages stored in: /home/gibdata8451/.ivy2/jars
org.apache.spark#spark-sql-kafka-0-10_2.12 added as a dependency
:: resolving dependencies :: org.apache.spark#spark-submit-parent-c3acf098-8db7-4a8a-bdd8-bfd295a805bf;1.0
        confs: [default]
        found org.apache.spark#spark-sql-kafka-0-10_2.12;3.4.1 in central
        found org.apache.spark#spark-token-provider-kafka-0-10_2.12;3.4.1 in central
        found org.apache.kafka#kafka-clients;3.3.2 in central
        found org.lz4#lz4-java;1.8.0 in central
        found org.xerial.snappy#snappy-java;1.1.10.1 in central
        found org.slf4j#slf4j-api;2.0.6 in central
        found org.apache.hadoop#hadoop-client-runtime;3.3.4 in central
        found org.apache.hadoop#hadoop-client-api;3.3.4 in central
        found commons-logging#commons-logging;1.1.3 in central
        found com.google.code.findbugs#jsr305;3.0.0 in central
        found org.apache.commons#commons-pool2;2.11.1 in central
:: resolution report :: resolve 866ms :: artifacts dl 32ms
        :: modules in use:
        com.google.code.findbugs#jsr305;3.0.0 from central in [default]
        commons-logging#commons-logging;1.1.3 from central in [default]
        org.apache.commons#commons-pool2;2.11.1 from central in [default]
        org.apache.hadoop#hadoop-client-api;3.3.4 from central in [default]
        org.apache.hadoop#hadoop-client-runtime;3.3.4 from central in [default]
        org.apache.kafka#kafka-clients;3.3.2 from central in [default]
        org.apache.spark#spark-sql-kafka-0-10_2.12;3.4.1 from central in [default]
        org.apache.spark#spark-token-provider-kafka-0-10_2.12;3.4.1 from central in [default]
        org.lz4#lz4-java;1.8.0 from central in [default]
        org.slf4j#slf4j-api;2.0.6 from central in [default]
        org.xerial.snappy#snappy-java;1.1.10.1 from central in [default]
        ---------------------------------------------------------------------
        |                  |            modules            ||   artifacts   |
        |       conf       | number| search|dwnlded|evicted|| number|dwnlded|
        ---------------------------------------------------------------------
        |      default     |   11  |   0   |   0   |   0   ||   11  |   0   |
        ---------------------------------------------------------------------
:: retrieving :: org.apache.spark#spark-submit-parent-c3acf098-8db7-4a8a-bdd8-bfd295a805bf
        confs: [default]
        0 artifacts copied, 11 already retrieved (0kB/18ms)
25/04/19 10:26:58 INFO SparkEnv: Registering MapOutputTracker
25/04/19 10:26:58 INFO SparkEnv: Registering BlockManagerMaster
25/04/19 10:26:58 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
25/04/19 10:26:58 INFO SparkEnv: Registering OutputCommitCoordinator
25/04/19 10:26:58 INFO MetricsConfig: Loaded properties from hadoop-metrics2.properties
```

Data Save at HDFS

SSH-in-browser

```
gibdata8451@cluster-e861-m:~/final_test$ hadoop fs -ls /BigData/news_data/
Found 7 items
drwxr-xr-x   - gibdata8451 hadoop          0 2025-04-19 10:39 /BigData/news_data/_spark_metadata
-rw-r--r--   1 gibdata8451 hadoop        128 2025-04-19 10:39 /BigData/news_data/part-00000-222e5d88-f203-49b5-aab8-b0ab9a164e54-c000.csv
-rw-r--r--   1 gibdata8451 hadoop          0 2025-04-19 10:27 /BigData/news_data/part-00000-53dbc3af-34fa-4c7a-9164-d29ee6dea0e5-c000.csv
-rw-r--r--   1 gibdata8451 hadoop        250 2025-04-19 10:39 /BigData/news_data/part-00000-5ba43b6c-5b00-4b1e-94e7-b7646eb155fc-c000.csv
-rw-r--r--   1 gibdata8451 hadoop          0 2025-04-19 10:38 /BigData/news_data/part-00000-f7adb00e-368d-4dd2-84a6-121d029539ba-c000.csv
-rw-r--r--   1 gibdata8451 hadoop         59 2025-04-19 10:39 /BigData/news_data/part-00001-a3862b77-fc93-41c2-a9d6-97e4d4b8009f-c000.csv
-rw-r--r--   1 gibdata8451 hadoop        124 2025-04-19 10:39 /BigData/news_data/part-00001-e9b1cb1d-9e13-4055-a475-eb11bf14fd42-c000.csv
```

Create Hive Table

```
CREATE EXTERNAL TABLE news_aggregated (
  window_start TIMESTAMP,
  window_end TIMESTAMP,
  source STRING,
  count INT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION '/BigData/news_data';
```

Query the Hive Table

```sql
SELECT * FROM news_aggregated LIMIT 10;
```

SSH-in-browser

```
hive> SELECT * FROM news_aggregated LIMIT 10;
OK
NULL    NULL    The Verge       52
NULL    NULL    Gizmodo.com     18
NULL    NULL    Wired   16
NULL    NULL    The Verge       52
NULL    NULL    The Verge       52
NULL    NULL    Gizmodo.com     18
NULL    NULL    Wired   16
NULL    NULL    Wired   16
NULL    NULL    Gizmodo.com     18
Time taken: 1.871 seconds, Fetched: 9 row(s)
```