Lab Subject - Velib Use case

Our use case is about Bike Sharing Systems, such as Vélib in Paris or Vélô in Toulouse. There is an API that allows to monitor bike stations in many cities, in France, in Europe and around the world: https://developer.jcdecaux.com

We will use this API to observe real-time rentals at each station. Ask for a API Key from https://developer.jcdecaux.com/#/signup.

If your API key is "XXX", you can check whether it works correctly by retrieving the list of all stations using the following command:

```
> curl https://api.jcdecaux.com/vls/v1/stations?apiKey=XXX
```

Try also : (with a Json « Prettifier »)

```
> curl https://api.jcdecaux.com/vls/v1/stations?apiKey=XXX | python -m json.tool
```

As we can see, the API provides us with the number of free slots ("available_bike_stands") in each station. If this number increases (respectively: decreases) in a station between two calls to the API, it means that bikes have been rented (respectively: returned) in the station.

The objective here is to set up an application that will display the evolution of the number of available slots, such as:

```
+1 MAZARGUES - ROND POINT DE MAZARGUES (OBELISQUE) (Marseille)
+14 Lower River Tce / Ellis St (Brisbane)
+2 2 RUE GATIEN ARNOULT (Toulouse)
+20 ANGLE ALEE ANDRE MURE ET QUAI ANTOINE RIBOUD (Lyon)
+14 Smithfield North (Dublin)
+28 52 RUE D'ENGHIEN / ANGLE RUE DU FAUBOURG POISSONIERE - 75010 PARIS
(Paris)
+6 RUE DES LILAS ANGLE BOULEVARD DU PORT - 95000 CERGY (Cergy-Pontoise)
+6 San Juan Bosco - Santiago Rusiñol (Valence)
+21 AVENIDA REINA MERCEDES - Aprox. Facultad de Informática (Seville)
+6 Savska cesta 1 (Ljubljana)
+31 DE BROUCKERE - PLACE DE BROUCKERE/DE BROUCKEREPLEIN (Bruxelles-
Capitale)
+7 BRICHERHAFF - AVENUE JF KENNEDY / RUE ALPHONSE WEICKER (Luxembourg)
...
```

To send a message/record, we must first create a topic. We create a topic " velib-stations " by executing the script kafka-topics.sh situated in the directory bin/ :

```
> ./bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic
velib-stations
```

Notice that we give to the command options the number of partitions and the replication rate of the topic. We can check that our topic "velib-stations" has been created by listing existing topics:

```
> ./bin/kafka-topics.sh --list --zookeeper localhost:2181
```

Velib Application,

[the scripts/application can be written in one of the following languages: Java, Scala, Python]

Practice Exercises :

1. Write a script: **ingest-data.py** that queries the Bike Sharing System API in a regular interval of time (say 10 seconds) and write the result json in a Kafka topic "*velib-stations"*.

2. Write a script: **stations-activity.py** that listens to the Kafka topic "velib-stations" that verifies the status of each station and in case of any change status change, sends the stations details to a new topic "*stations-status"*.

3. Write a script: **empty-stations.py** that listens to the Kafka topic "*stations-status*" and does following tasks:
- send a message to a topic "*empty-stations*" as soon as a station becomes empty (when it was not empty before).
- send a message to the "*empty-stations*" topic as soon as a station is no longer empty (when it was empty before).

4. Write a script: **alert-empty-stations.py** that prints a message in the console as soon as a station becomes empty (whereas it was not empty before). This message should contain:
- the address of the station,
- the city of the station.

5. Write a script: **archive-data.py** that archives the input data in the topic "*velib-stations*" in a text file.

6. Write a script: **monitor-kafka.py** that monitors all the Kafka topics and prints the status of each Kafka topic in the console in real-time for monitoring purposes. The status information should be in the format: *Topic-name, Partition-id, offset-id, timestamp.*

WARNING! Plagiarism will not be tolerated and everyone who are involved will be strictly penalised.

You need to upload a lab-scripts.zip file on the moodle containing all the scripts (without archive text files). Number of students per group 3. **Deadline : Decembre 25th, 2023.**

Note :
This lab is adapted from the MOOC "Gérez des flux de données temps reel", Open Class Room
https://openclassrooms.com/fr/courses/4451251-gerez-des-flux-de-donnees-temps-reel?status=published