# Mini projet Spark

Contexte : Vous travaillez pour la métropole de Paris. Cette dernière souhaite augmenter l'usage de son service de vélos partagés (vélib) et améliorer l'expérience de ses abonnements.

Cela passera notamment par déterminer si les capacités des stations sont correctement dimensionnées.

Pour cela elle souhaite miser sur la data pour améliorer son service. Vous avez été mandaté pour mettre en place un système de traitement de données.

Pour cela vous utiliserez l'API open data permettant à tout le monde de récupérer ces données (cf lien).

**Partie 1 : Récupération des données**

1. Se rendre sur https://data.opendatasoft.com/pages/home/ et rechercher "velib temps réel". Et aller sur l'onglet API. Voir ce que renvoie l'API. Identifier les champs qui pourraient vous intéresser.

2. Récupérer l'url utilisé pour envoyer la requête. Avec python, faire en sorte de faire une requête avec python pour récupérer les données d'une station. Récupérer le résultat sous forme de dictionnaire.

3. Faire une fonction get_velib_data(nrows) qui permet de récupérer nrows lignes de données.

4. Créer un producer kafka qui sera charger d'envoyer les données reçu sur une instance Kafka. Faire en sorte de faire un script qui récupère toutes les minutes des données de l'API et l'envoie via le producer.

5. Bonus : vérifier que vous ne récupérez pas deux fois la même donnée.

**Partie 2 - Traitement en ligne**

On souhaite désormais faire du traitement en ligne pour mesurer statistiques d'utilisation des stations en temps réel.

Faire en sorte de connecter Spark Streaming à Kafka et calculer la moyenne pour chaque station disponible :

- du nombre de vélos mécaniques disponibles
- du nombre de vélos électriques disponibles
- du nombre de place libres disponibles

Vous pourrez également tenter de calculer la moyenne d'occupation des stations dans une zone géographique proche ou toute autre métrique pourrait donner une meilleur compréhension de l'usage des stations.

Bonus : Stocker le résultat des traitements dans un fichier ou une base de données.

**Partie 3 - Traitement par batch**

On veut également faire des traitements sur des batchs afin d'avoir des statistiques d'utilisation à plus long terme.

1. Faire un consumer qui sera chargé d'agréger l'ensemble des données récupérée dans un fichier csv.

2. Lancer le consumer et récupérer un premier fichier avec des données. Vous pouvez laisser tourner le consumer afin de collecter davantage de données.

3. Se rendre sur https://community.cloud.databricks.com qui permet d'avoir accès à un petit cluster spark gratuitement.

Lancer un cluster (toujorus gratuitement) et lancer un notebook.

Charger le fichier csv dans databricks.

Charger dans un dataframe le fichier csv. Si il est gros, en prendre un sous ensemble dans un premier temps. Cela facilitera le débugage.

4. Afficher le nombre de partition sur lequel le jeu de donnée est stocké

5. Analyser les données au regard de la problématique soulevée : comment améliorer l'expérience utilisateur et renforcer l'usage.

Parmi les analyse qui pourront être menées sera :

- le nombre moyen, le min et le max de place de velib pour chaque station
- Le nombre moyen de place de velib pour chaque zone (à définir vous même)
- Le nombre de station et / ou la liste des stations qui ont parfois aucun vélib disponible

Faire en sorte de refaire les mêmes traitement avec l'API SQL

Bonus : Faire en sorte que le jeu de donnée ne soit stockée sur une qu'une seule partition. Comparer le temps de traitement entre une partition et deux partitions

**Partie 4 - Bonus : Machine learning**

Récupérer toutes les données d'une station seulement. Avec Spark ml, faites un modèle qui aura pour but de prédire si une station sera remplie à un instant t*t* en fonction du nombre de place disponibles t_1*t*1, t_2*t*2 t_3*t*3.