Une image contenant texte, Police, Graphique, calligraphie

Description générée automatiquement

**BUT 3 INFORMATIQUE**

**Parcours A FI**

**Réalisations d'applications**

**RAPPORT TP3**

**Analyse de mots**

**Prénom et Nom de l’étudiant :** Akash Selvaratnam

**Groupe :** 303

**Promotion :** 2023-2024

# Sommaire

Table des matières

[Sommaire 2](#_Toc158280491)

[Étape 1 – « Vive les petits producteurs » 3](#_Toc158280492)

[Étape 2 – Star de Cinéma : « 26l / 100 km » - Qui suis-je ? 4](#_Toc158280493)

[Étape 3 –« Are you Redis »? 5](#_Toc158280494)

# Étape 1 – « Vive les petits producteurs »

Pour l’étape 1, j’ai crée une nouvelle fonction getNumberWorld() dans le fichier config.js permettant de retourner le nombre de mots qui se trouve dans le fichier .env.

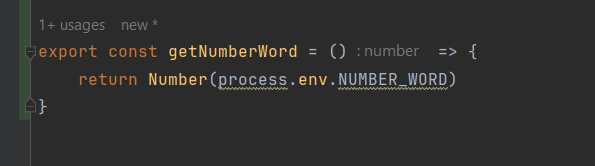


Figure 1 : Fonction permettant de retourner le nombre de mots

J’ai ensuite ajouter les configurations nécessaire dans le fichier .env pour la configuration de redPanda en fonction des valeurs présent dans le readfile

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

Figure 2 : Configuration du serveur redPanda

J’ai construit mon conteneur avec la commande suivante docker build --tag red-panda-producer . et j’ai lancer mon conteneur avec la commande suivante docker run -d --name RedPanda-Producer --volume=C:/TP\_NODEJS/TP3/prod-red-panda/.env:/home/node/app/.env --network=redpanda-quickstart\_redpanda\_network red-panda-producer.

J’ai pu créer mon topic avec mes enregistrement.



Figure 3 : Console de redPanda

On peut également visualiser mes enregistrement sur l’interface web de redPanda.

Une image contenant texte, capture d’écran, Police, document

Description générée automatiquement

Figure 4 : Interface redpanda

# Étape 2 – Star de Cinéma : « 26l / 100 km » - Qui suis-je ?

Pour l’étape 2, j’ai me suis connecté au serveur redpanda au port 19092.

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

Figure 5 : Configuration redpanda

Ensuite, j’ai créé une fonction consommeur() qui permet de se connecter à mon topic et d’afficher l’ensemble des enregistrement de ces topic.



Figure 6 : Fonction Consommeur()

J’ai également réaliser une fonction getDate() permettant de transformer le timestamp sous format de date et de changer le format de la date avec le format suivant dd/mm/aaaa à hh : mm.

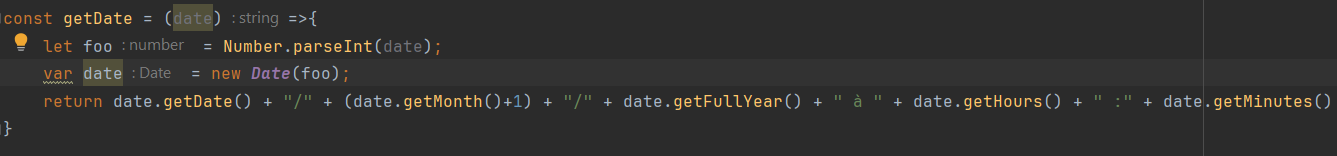


Figure 7 : Fonction getDate()

On peut ensuite lancer le serveur afin d’apercevoir le message que nous affiche la console.

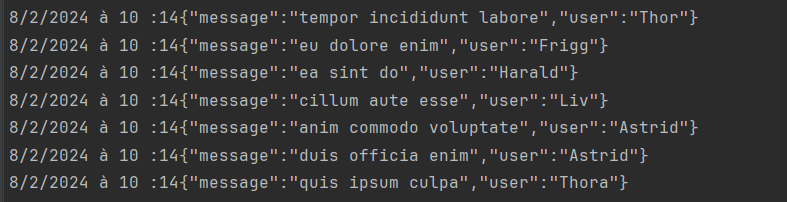


Figure 8 : Résultat console de la fonction consommateur

# Étape 3 –« Are you Redis »?

Pour l’étape 3, j’ai découper le message qu’obtient le consommateur et je le découpe sous format de mot afin de pouvoir exécuter la fonction incr de la bibliothèque redis sur chacun des mots que le consommateur obtient.

Tout d’abord, il faut se connecter a notre serveur redis en indiquant le host, le port et le mot de passe.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Figure 9 : Connexion au serveur redis

Ensuite dans la fonction connexion du Consommer, je transforme le message sous format json afin de pouvoir découper ce message sous format de mot grâce à la fonction split() et pour chaque mot, j’utilise la fonction incr pour pouvoir incrémenter l’occurrence des mot (clé).

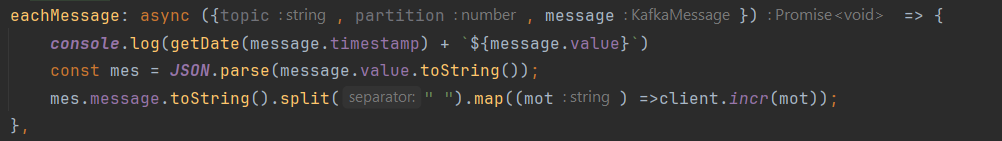


Figure 10 : Incrémentation de l'occurrence des mots

On peut ensuite, lancer le serveur et consulter les résultats de la fonction consommeur() sur le site web React.

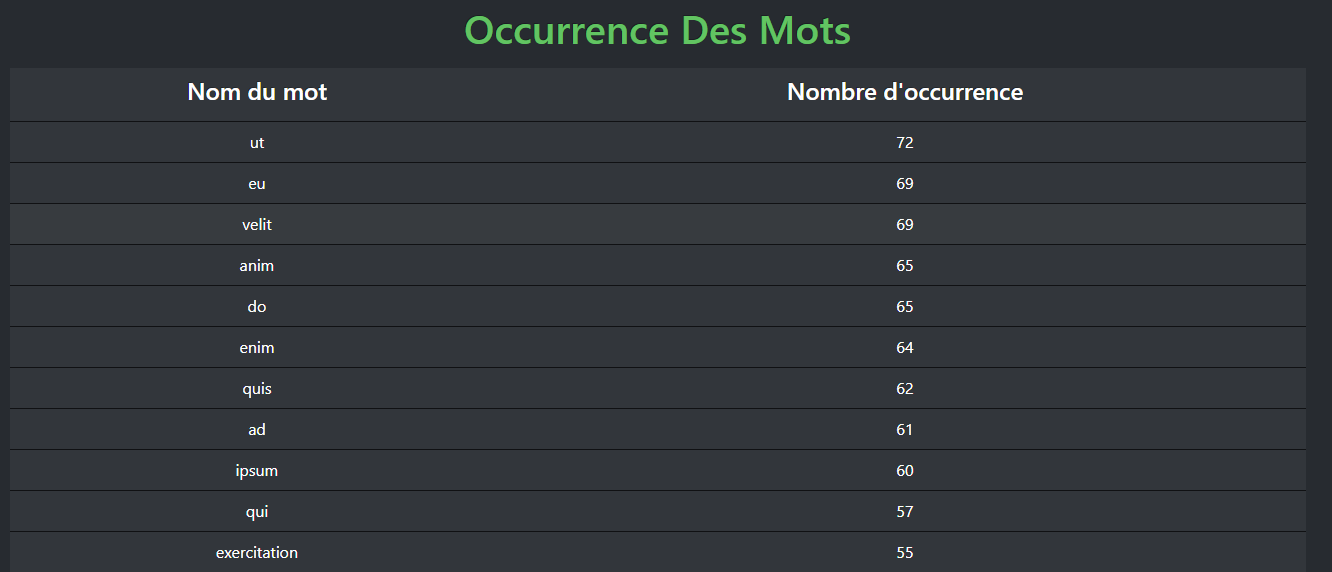


Figure 11 : Page React (pour pouvoir consulter les résultats de l'opération INCR de redis)