

Compte rendu - Sujet 3

Analyse de mots

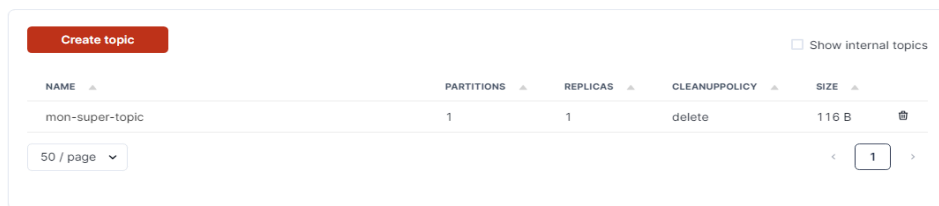
Étape 1 - « Vive les petits producteurs »

Pour cette étape, j'ai commencé par écrire la méthode `getNumberWord()` qui retourne la valeur `NUMBER_WORLD` du fichier `.env` (ou renvoie 1 comme valeur par défaut pour ne pas rien envoyer en cas d'erreur) :

```
export const getNumberWord = () => {  
  return Number(process.env.NUMBER_WORLD) || 1;  
}
```

Après je l'utilise comme paramètre dans la méthode `getStringMessage()` dans le `server.js`.

Pour l'étape suivante j'ai créé le topic `mon-super-topic` (comme dans le fichier `.env`)



Maintenant quand je lance le projet ça me génère des phrase sur mon topic que je peux voir directement sur le panel :

```
{ topic: 'mon-super-topic', user: 'Liv', message: 'veniam duis id' }  
{ topic: 'mon-super-topic', user: 'Thora', message: 'sunt in anim' }
```

Messages		Consumers	Partitions	Configuration	ACL	Documentation
PARTITION		START OFFSET	MAX RESULTS	FILTER		
All		Newest - 50	50	Actions		
OFFSET		PARTITION	TIMESTAMP	KEY	VALUE	Preview
0		0	08/02/2024 15:48:48	⌚	{"message":"nostrud veniam ullamco","user":...	
1		0	08/02/2024 15:48:48	⌚	{"message":"occaecat quis eiusmod","user":..."}	

À cette étape, j'ai créé l'image producteur mais quand j'essaie de la run sur le conteneur, je n'y arrive pas je reçois une erreur liée à l'impossibilité de connexion au broker Kafka.

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

Pour cette étape j'ai commencé par créer un nouveau projet `cons-red-panda`. Ensuite, je suis allé voir la documentation sur [kafka.org](https://kafka.apache.org/) (avec la fonction `consumer` en bas de la page). J'ai repris les bouts de code que j'ai adapté pour mon application. Cette méthode `connexion()` créer un consumer qui s'abonne à un topic et qui est retourné.

```
async function connexion() : Promise<...> {
  try {
    const kafka : Kafka = new Kafka( config: {
      clientId: 'my-app',
      brokers: ['172.18.0.2:9092'],
    });

    const consumer : Consumer = kafka.consumer( config: { groupId: 'group' });

    await consumer.connect();
    console.log('connecté au broker RedPanda');

    await consumer.subscribe( subscription: { topic: 'mon-super-topic' });
    console.log('abonné au topic mon-super-topic');

    return consumer;
  } catch (error) {
    console.error('erreur de connexion au broker:', error);
  }
}
```