MCR Génération

1. Description

Cette application est une API codé en FastAPI qui permet de générer un compte rendu de réunion à partir de la transcription de cette réunion. Elle est l'une des briques d'une architecture microservice du projet **Mon Compte Rendu**.

2. Fonctionnalités

- **Réception d'une transcription diarisée** : L'application reçoit une transcription diarisée (i.e. chaque segment de la transcription est rattaché à un locuteur) par segment.
- **Génération d'un compte rendu** : Selon le type de réunion un compte rendu est généré par une chaîne d'appels succéssifs à un LLM.
- **Classification des décisions par thème** : Lorsque le type de compte-rendu est un relevé de décisions une classification est opérée pour associer chaque décision à un thème abordé dans l'ordre du jour.

3. Prérequis

Avant de démarrer, assurez-vous d'avoir installé les éléments suivants sur votre machine :

- Docker
- Docker Compose

Il faut également avoir capté la transcription d'une réunion avec le module mcr-speech-transcription.

4. Installation et Lancement du Module en local

Comme susmentionné, l'api est une brique d'une architecture micro-service qui nécessitent d'autres briques que sont : - mcr-core : elle s'occupe de capter l'audio d'une plateforme de visioconférence par chunks de x secondes et d'envoyer dans un topic kafka - mcr-gateway : elle joue le rôle orchestrateur des différentes briques de l'architecture - mcr-speech-transcription : elle permet de transcrire ce qui est dit dans une réunion.

4.1. Clôner les dépôts des différents microservices

Commencez par cloner ce dépôt sur votre machine locale et ainsi que les dépôts des autres composants :

```
git clone git@github.com:IA-Generative/mcr-generation.git
git clone git@github.com:IA-Generative/mcr-gateway.git
git clone git@github.com:IA-Generative/mcr-core.git
git clone git@github.com:IA-Generative/mcr-speech-transcription.git
```

4.2. Lancer le fichier docker-compose disponible à la racine

Pour créer le réseau, exécutez la commande suivante :

```
docker compose -f main-docker-compose.yaml up
```

L'API core initilisera les tables au niveau de la base de données

4.2. Lancer les commandes suivantes pour créer et démarrer une réunion

- *Créer une réunion*

```
curl --location 'http://localhost:8001/api/meetings' \
--header 'Content-Type: application/json' \
--data '{
    "name":"ma reunion 2",
    "url":"<URL REUNION COMU>",
    "name_platform":"COMU",
    "meeting_type":"instance",
    "start_date":"2024-10-22",
    "end_date":"2024-10-22",
    "meeting_agenda":"ordre du jour",
    "user_id":1
}'
```

NB : L'url de la réunion doit avoir le format suivant : *https://webconf.comu.gouv.fr/en-US/meeting/<**ID DU MEETING**>?secret=<**SECRET**>*

- *Lancer la transcription d'une réunion*

```
curl --location 'http://localhost:8001/api/meetings/1/start-transcription'
```

Une fois la réunion lancé un bot se connecte à une réunion COMU, puis récupère l'audio et l'envoie dans le topic kafka et enfin le module transcription prend la main en asynchrone pour consommer et traiter les messages de la queue kafka.

- *Stopper la réunion*

```
curl --location 'http://localhost:8001/api/meetings/1/stop-transcription'
```

- *Générer le compte rendu d'une réunoon*

```
curl --location 'http://localhost:8000/api/meetings/1/report'
```

- *Supprimer les composants*

```
docker compose -f main-docker-compose.yaml down
docker rm $(docker ps -a -q)
```