

TP1 : Simulateur de trames télématiques

Énoncé

Ce TP est à faire en groupe de 2.

Écrire un programme qui va générer les trames MQTT décrites dans le cahier des charges. Ce programme doit être capable de générer les trames de plusieurs véhicules avec des données aléatoires (au minimum pour le nombre de passagers) respectant les données d'entrées. Ces informations doivent être récupérées depuis le fichiers de configuration json « ssie_vehicles_data.json » disponible dans moodle.

Le fichier de configuration doit être lu en utilisant un chemin relatif.

Vous implémenterez une IHM permettant de configurer et manipuler ces données de simulation. L'IHM doit par exemple pouvoir :

- Modifier la fréquence d'envoi, les informations de connexions
- Sélectionner/Désélectionner des véhicules
- Ajouter/Modifier/Supprimer des véhicules
- Arrêter/Démarrer la simulation
- Autres idées que vous jugerez intéressantes pour améliorer la simulation

Le programme doit être capable de générer les trames directement en ligne de commande (ou via l'IHM).

Le programme doit aussi avoir en option la possibilité de lancer un processus qui va consommer les messages MQTT produits.

Il est attendu que le code source soit découpé en classe ou (au moins) en fonctions, et commenté.

Le fichier README.md doit contenir les sections suivantes :

- Les lignes de commandes pour la compilation et l'exécution du programme
- Une présentation de l'architecture du programme
- Un manuel utilisateur pour l'IHM.

Données d'entrées :

- Langage à utiliser : au choix
- Cahier des charges disponible sur moodle (Cloud_2023_2_CDC_Exercice.pdf).
- Fichier de données disponible sur moodle (fichiers configuration json « ssie_vehicles_data.json »).

Prérequis :

- Docker : un conteneur EMQX s'exécutant en local sur Docker.
- Cours sur MQTT avec commande pour exécuter l'image docker EMQX

Objectif

- Manipuler le broker MQTT EMQX
- Implémenter d'un programme produisant des messages MQTT à destination du broker avec une IHM.

Livraison attendue

Le template de l'arborescence attendue pour les exercices est disponible ici :

https://gitlab.cri.epita.fr/mathias.choquet/ssie_cloud_project_template

Le rendu se fait via le dépôt du code source dans le répertoire « vehicle_data_producer ». Le résultat doit être visible dans EMQX.

Grille de notation (sur 20 points)

- Fonctionnement : 6 points
- IHM : 4 points
- Option de consommation des trames MQTT : 2 points
- Conception du code source (découpage en classe ou en fonction) : 2 points
- Qualité du code source (optimisation, factorisation, ré-utilisabilité) : 2 points
- Gestion des erreurs ou des exceptions : 2 points
- Qualité du fichier README.md : 2 points