

# Le projet GraphHopper

Romain Tavenard

## 1 Organisation

- Groupes de 2 étudiants **du même groupe de TD**
- Déclaration des groupes :
  - au plus tard le jeudi 21/11/2019, pendant la séance de TD
  - si pas inscrit dans un groupe à cette date : note de 0 au projet
- Rendu final : au plus tard le vendredi 13/12/2019, 23h59

## 2 Énoncé

Dans ce projet, vous allez permettre à un utilisateur, étant donné un ensemble de points d'intérêt, de retrouver la liste de tous les bus du réseau STAR qui, à l'instant présent, se trouvent à moins de 500m (par la route, pas à vol d'oiseau) d'au moins un de ces lieux. Pour chacun des bus concernés, vous devrez afficher :

- le numéro de la ligne,
- la direction,
- le point d'intérêt dont il est à moins de 500m,
- la distance à ce point d'intérêt.

## 3 Données à votre disposition

Votre programme prendra en entrée des données issues de 4 sources différentes :

- a. Un fichier `targets.csv` de la forme :

```
INTITULE;  
Rennes, République;  
Rennes, Villejean;  
10 Rue de Saint Malo, Rennes  
[...]
```

Ce fichier définit la liste des points d'intérêts autour desquels on souhaite repérer des bus.

- b. Un fichier `credentials.json` qui stocke votre clé d'API GraphHopper, comme vu en TD.
- c. L'[API STAR](#) qui permet d'accéder en temps réel à la position des bus en circulation (sans avoir besoin de créer de clé d'API, pour une fois).
- d. L'API GraphHopper qui vous servira à calculer des distances.

Pour votre rendu, **vous devrez vous-mêmes créer des fichiers de test** qui soient pertinents pour le problème étudié et évaluer votre programme sur ces fichiers (toutefois, vous ne devrez pas déposer le fichier `credentials.json` car il contient vos identifiants personnels).

Pour accéder aux données de position des bus, vous interrogerez l'API STAR via une requête HTTP. Pour ce qui est de l'API GraphHopper, vous utiliserez [le module `graphh`](#).

## 4 Évaluation

Pour l'évaluation, il sera porté une attention particulière à ce que votre code soit lisible (commentaires : ni trop ni trop peu, nom de variables / fonctions pertinents, etc.) et bien découpé en blocs comme vous avez pu le voir lors du TD "Analyse et décomposition de problèmes".