

# API REST d'accès aux données

Xavier André & Romain Tavenard

## 1 Rappel : organisation de votre code

Pour ce TD, vous créerez un nouveau fichier `td_api_rest.py`. Dans ce fichier, votre code sera organisé de la manière suivante :

```
# Imports
import requests
from graphhopper import GraphHopper

# Fonctions
def [...]

# Tests
[...]
```

Notamment, vous définirez vos fonctions en début de fichier et les appels seront listés en fin de fichier. De cette manière, vous pourrez, d'une question à l'autre, réutiliser les fonctions déjà codées au besoin.

## 2 Énoncé

1. Se rendre sur le [site de la STAR](#) et trouver l'API indiquant la position des bus de la STAR en temps réel. Cliquez sur l'onglet "API" pour accéder aux options de requête. Essayez notamment d'ajouter le *facet* `etat` et observez les valeurs qu'il peut prendre.
2. Combien de résultats (`records`) retourne la requête proposée par défaut ? Et combien de résultats sont contenus dans la base (attribut `hits`) ? Comment faire pour vous assurer d'avoir tous les résultats lorsque vous formulez une requête ?

Pour les questions suivantes, il est conseillé d'importer le module `pprint` qui permet d'afficher de manière claire les dictionnaires :

```
from pprint import pprint

[...]
pprint(mon_joli_dictionnaire)
```

3. Écrire une fonction qui retourne une liste des positions des bus **en service** sur le réseau, chaque bus étant représenté par un dictionnaire qui contient les clés `numerosbus`, `nomcourtligne`, `destination` et `ecartsecondes`. Pour cela, consultez l'interface d'édition de requêtes de l'API de la STAR (celle que vous avez trouvée à la question 1) et ajoutez les attributs souhaités à la liste des *facets*, puis notez l'URL générée (clic droit sur le lien du bas de la page, puis "Copier le lien").
4. Écrire une fonction qui retourne le nombre de bus en avance (qui ont un attribut `ecartsecondes` positif) dans le jeu de données.
5. Écrire une fonction qui retourne, pour chaque ligne de bus, une liste des `ecartsecondes` des bus de cette ligne.

### 3 Utilisation de l'API GraphHopper

1. Se créer un compte sur l'API GraphHopper et générer une clef d'API pour ce compte.
2. Stocker cette clé d'API dans l'attribut `"clef_GH"` d'un fichier `credentials.json` stocké dans votre dossier `data`. Le fichier aura la structure suivante :

```
{
    "clef_GH": "VOTRE_CLE_ICI"
}
```

3. Dans votre script Python, lisez cette clef d'API depuis le fichier en question puis générez un objet client GraphHopper : `gh_client = GraphHopper(api_key=YOUR_API_KEY)`
4. En utilisant la fonction `latlong_to_address` du module `graphh` et les fonctions créées plus haut dans ce TD, écrivez une fonction qui prend en entrée un numéro de bus et un client GraphHopper et retourne l'adresse où se trouve le bus en question.