

API web d'accès aux données

Planche de TD pour un cours dispensé à l'université de Rennes 2

1 Principe

Cet énoncé de TD va vous amener à effectuer des requêtes aux APIs que vous avez déjà manipulées dans les TD précédents, sous la forme de quelques mini-projets. On utilisera notamment le module GraphHopper [lien vers l'aide du module graphh](#).

1.1 Quel covoiturage ?

Dans cette partie, on suppose que l'on cherche à déterminer, parmi plusieurs choix possibles, un trajet optimal avec les contraintes suivantes :

- les points de départ et d'arrivée sont fixés
- on doit récupérer, sur le trajet, une personne en covoiturage et l'on peut choisir entre plusieurs personnes situées à des positions différentes.

La question peut par exemple être posée sous la forme suivante :

Lors d'un trajet Rennes - Marseille, vaut-il mieux (en termes de temps de trajet) prendre quelqu'un à "Paris 14ème arrondissement", "Lyon 1er arrondissement" ou "Bordeaux" ?

1. Écrivez une fonction qui prend en entrée :
 - un lieu d'origine (sous la forme d'une chaîne de caractères)
 - une destination (sous la forme d'une chaîne de caractères)
 - une liste d'options sur le trajet (sous la forme d'une liste de chaînes de caractères)
 - un client d'API GraphHopperet retourne l'option correspondant au trajet le plus court (en temps). En cas d'égalité, votre fonction retournera l'une des options correspondant au trajet le plus court (en temps).

1.2 Où se retrouver ?

Plusieurs ami(e)s souhaitent se retrouver, tout en minimisant leur temps de trajet. Plus précisément, ils souhaitent que **la somme des distances parcourues par eux tous pour se retrouver soit la plus faible possible**.

La question peut par exemple être posée sous la forme suivante :

Pour trois amis habitant respectivement Paris, Auxerre et Lyon, est-il préférable (selon le critère énoncé ci-dessus en gras) de se rencontrer à Rennes ou à Strasbourg ?

2. Écrivez une fonction qui prend en entrée :
 - une liste de positions des amis (chaque position étant représentée par une chaîne de caractères)
 - une liste de lieux de rencontres possibles (sous la forme d’une liste de chaînes de caractères)
 - un client d’API GraphHopperet retourne l’option la plus favorable (par rapport au critère énoncé en gras plus haut). En cas d’égalité, votre fonction retournera l’une des options les plus favorables (par rapport au critère énoncé en gras plus haut).

1.3 Bonus : Trouver un partenaire sportif

Ici, on suppose que l’on souhaite faire du sport et que, pour un sport donné, on veut chercher dans une liste de partenaires potentiels celui ou celle qui se trouve le plus proche de nous.

La question peut par exemple être posée sous la forme suivante :

- Voici ma liste de contacts :
 - Pauline, joue au tennis, se trouve “Place du recteur Henri Le Moal, Rennes, France”
 - Ernest, joue au football, se trouve “Place du Parlement de Bretagne, Rennes, France”
 - Felix, joue au tennis, se trouve “Rue Lebastard, Rennes, France”
 - Sarah, joue au football, se trouve “Place du Parlement de Bretagne, Rennes, France”
 - Ingrid, pratique la course à pied, se trouve “Mail François Mitterrand, Rennes, France”
 - Sachant que je me trouve Place de la République à Rennes, quel est, dans ma liste de contacts, celui ou celle qui se trouve le plus proche de moi et qui joue au tennis ?
3. Écrivez une fonction qui prend en entrée :
 - une position (représentée par une chaîne de caractères)

- une liste de contacts (sous la forme d'une liste de dictionnaires telle que `liste_dicos` ci-dessous)
 - une activité physique (représentée par une chaîne de caractères)
 - un client d'API GraphHopper
- et retourne le nom de la personne dont la localisation est la plus proche de nous et qui pratique le sport voulu. En cas d'égalité, votre fonction retournera le nom de l'une des personnes dont la localisation est la plus proche de nous et qui pratique le sport voulu.

```
liste_dicos = [  
    {  
        "nom": "Pauline",  
        "sport": "Tennis",  
        "localisation": "Place du recteur Henri Le Moal, Rennes, France"  
    },  
    {  
        "nom": "Ernest",  
        "sport": "Football",  
        "localisation": "Place du Parlement de Bretagne, Rennes, France"  
    },  
    {  
        "nom": "Felix",  
        "sport": "Tennis",  
        "localisation": "Rue Lebastard, Rennes, France"  
    },  
    {  
        "nom": "Sarah",  
        "sport": "Football",  
        "localisation": "Place du Parlement de Bretagne, Rennes, France"  
    },  
    {  
        "nom": "Ingrid",  
        "sport": "Course à pied",  
        "localisation": "Mail François Mitterrand, Rennes, France"  
    }  
]
```