

Activité 13c : L'Open Data, des données accessibles à tous

Généralement les banques de données OpenData mettent à disposition un certains nombres de données accessibles directement à l'aide d'une requête transmise sur le serveur. Cette requête interrogera alors l'API "Application Programming Interface" de la banque de données qui renverra la réponse dans un format connu. Ces API assurent ainsi la diffusion de données dynamiques qui pourront comporter des données mis à jour en temps réel.

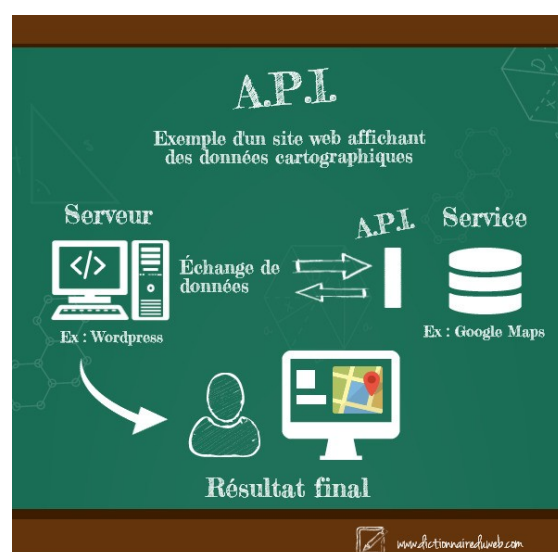
Comment interroger un API ? Comment décoder et traiter les données transmises par l'API ?

1. L'API

Définition

Une API est une interface de programmation qui permet de se « brancher » sur une application pour échanger des données. Une API est ouverte et proposée par le propriétaire du programme.

Ce mécanisme améliore l'interopérabilité puisque que le dialogue s'effectue avec des règles ouvertes, simples et indépendantes du matériel utilisé. En effet l'interrogation de l'API est réalisée via une requête *http* classique et la réponse sera aussi portée par une trame *http* dans laquelle les données seront organisées selon un format connu et ouvert (XML, JSON, KML...)



Fonctionnement de l'API

Pour exemple, la requête *http* suivante permet d'obtenir le jeu de données au format JSON portant sur les défibrillateurs localisés sur la ville de Bordeaux

<http://odata.bordeaux.fr/v1/databordeaux/defibrillateurs/?format=json>

1. **Envoyer** cette requête à l'API à partir du navigateur Firefox puis **coller** en dessous la réponse reçue avec un affichage en données brutes
2. **Visualiser** ces données sous le format d'affichage JSON puis **lister** le nom des informations données pour chaque borne.

2. Le format JSON

JavaScript Object Notation (JSON) est un format de données textuelles dérivé de la notation des objets du langage JavaScript. Il permet de représenter de l'information structurée comme le permet XML par exemple.

Le principal avantage de JSON est qu'il est simple à mettre en œuvre par un développeur tout en étant complet, il est aussi peu verbeux, ce qui le rend lisible aussi bien par un humain que par une machine et ses types de données sont connus et simples à décrire.



Un document JSON a pour fonction de représenter de l'information accompagnée d'étiquettes permettant d'en interpréter les divers éléments. Dans notre cas il est composé d'ensembles de paires "nom" (alias "clé") / "valeur"

3. Indiquer la clé et la valeur de l'adresse du lieu où est situé le défibrillateur N°1.

3. Exploitation de données JSON sous Python

Pour pouvoir interroger la base de donnée, il faut venir questionner l'API à l'aide d'une adresse internet (ou URL) comme nous l'avons vu précédemment.

La requête s'effectue en une seule instruction Python avec la méthode `requests.get()`. Cette dernière reçoit la réponse de l'API au format JSON qu'il faut arranger avec la méthode `resp.json()`. Cette méthode renvoie la liste des données sous la forme d'un dictionnaire python.

Reprenons notre exemple des données répertoriant les défibrillateurs installés dans la région Bordelaise. Ce jeu de données est en fait un dictionnaire contenant une liste de dictionnaires. Le programme suivant permet d'afficher la liste des adresses où sont implantés les machines.

```
import json, requests

url = 'http://odata.bordeaux.fr/v1/databordeaux/defibrillateurs/?format=json'
resp = requests.get(url, verify=True)
dico = resp.json()
for key in dico.keys():
    donnees=dico.get(key)
    for i in range(len(donnees)):
        print(donnees[i].get("adresse"))
```

4. Rappeler le rôle des instructions `data.keys()` et `data.get()`

5. Saisir ce code sur un IDE Python puis l'exécuter. **Commenter** ce programme en repérant particulièrement la boucle qui balaye le dictionnaire général puis celle qui permet de balayer la liste de dictionnaires. *Pour comprendre le programme il est conseillé de le modifier en changeant les noms de clé et en affichant des variables avec l'instruction `print()`*

x **Créer** et **tester** un programme capable d'indiquer le nombre de défibrillateurs de type DAE installés à Bordeaux.

Pour aller plus loin

x **Créer** et **tester** un programme capable de donner les adresses des 6 défibrillateurs les plus proches (classés par ordre croissant) à partir d'une position donnée par l'utilisateur (longitude et latitude).

Pour résoudre ce problème, il est sans doute nécessaire de mettre en application vos connaissances sur les vecteurs et notamment les calculs de module.

La position (x=-0,56, y=44,86) doit renvoyer la liste de défibrillateurs suivant :

boulevard Alfred Daney
Quai des Chartrons
87 boulevard Alfred Daney
quai de Queyries
29, rue Camille Godard
esplanade Linné

