Collecte Automatisée de Données Web

Séance 1 de cours/TD

IUT SD Dole

Objectifs:

- Introduction à la collecte de données Web avec des API
- —
- —

API : Qu'est-ce que c'est ? [Cours]

Une Application Programming Interface (ou API) est une interface de programmation qui permet d'utiliser une application tiers dans un programme ou une application que vous développez.

D'un point de vue informatique, une API est une porte d'entrée clairement identifiée par laquelle un logiciel offre des services à d'autres logiciels (ou utilisateurs). L'objectif d'une API est de fournir un point d'accès à une fonctionnalité qui soit facile à utiliser et qui masque les détails de la mise en oeuvre. Par exemple, l'API Sirene permet de récupérer la raison sociale d'une entreprise à partir de son identifiant Siren en interrogeant le référentiel disponible sur Internet directement depuis un script python ou R, sans avoir à connaître tous les détails du répertoire Sirene.
Cela peut se rapprocher de la notion de bibliothèque/package, mais avec la différence majeure que l'API repose sur le contact avec quelque chose d'extérieur (un programme, une base de données, un serveur,...) là où les bibliothèques sont un ajout de code directement accessible à votre programme.

On va s'interesser plus spécifiquement dans ce cours aux API de type REST (pour representational state transfer) qui permettent à un site web de donner à des tiers la possibilité de manipuler directement sa base de donnée sous-jacentes. Une API REST basée sur le protocole web HTTP est définie par :

- un URI de base, comme http ://api.example.com/collection/ ; C'est l'adresse à laquelle se connecter pour atteindre les données.
- des méthodes HTTP standards (par ex. : GET, POST, PUT, PATCH et DELETE) ; Il s'agit des opérations possibles sur les données. On utilisera principalement GET, les autres servant à effectuer des modifications sur la base.

Les API présentent de multiples avantages :

- Les API rendent les programmes plus reproductibles. En effet, grâce aux API, il est possible de mettre à jour facilement les données utilisées par un programme si celles-ci évoluent. Cette flexibilité accrue pour l'utilisateur évite au producteur de données d'avoir à réaliser de multiples extractions, et réduit le problème de la coexistence de versions différentes des données.
- Grâce aux API, l'utilisateur peut extraire facilement une petite partie d'une base de données plus conséquente.
- Les API permettent de mettre à disposition des données tout en limitant le nombre de personnes ayant accès aux bases de données elles-mêmes.
- Grâce aux API, il est possible de proposer des services sur mesure pour les utilisateurs (par exemple, un accès spécifique pour les gros utilisateurs).

Utiliser une API en Python [Cours]

Le mode principal de consultation d'une API consiste à adresser une requête à cette API via un logiciel adapté (R, Python, Java...). Comme pour l'utilisation d'une fonction, l'appel d'une API comprend des paramètres qui sont détaillées dans la documentation de l'API.

Voici les éléments importants à avoir en tête sur les requêtes :

- Le point d'entrée d'un service offert par une API se présente sous la forme d'une URL (adresse web). Chaque service proposé par une API a sa propre URL. Par exemple, dans le cas de l'OpenFood Facts, l'URL à utiliser pour obtenir des informations sur un produit particulier (l'identifiant 737628064502) est https://world.openfoodfacts.org/api/v0/product/737628064502.json

- Cette URL doit être complétée avec différents paramètres qui précisent la requête (par exemple l'identifiant Siren). Ces paramètres viennent s'ajouter à l'URL, souvent à la suite de ?. Chaque service proposé par une API a ses propres paramètres, détaillés dans la documentation.

- Lorsque l'utilisateur soumet sa requête, l'API lui renvoie une réponse structurée contenant l'ensemble des informations demandées. Le résultat envoyé par une API est majoritairement aux formats JSON ou XML (deux formats dans lesquels les informations sont hiérarchisées de manière emboitée). Plus rarement, certains services proposent une information sous forme plate (de type csv).

Du fait de la dimension hiérarchique des formats JSON ou XML, le résultat n'est pas toujours facile à récupérer mais python propose d'excellents outils pour cela. Certaines bibliothèques, notamment `json` ou `pandas`, facilitent l'extraction de champs d'une sortie d'API.
Dans certains cas, des packages spécifiques à une API ont été créés pour simplifier l'écriture d'une requête ou la récupération du résultat. Par exemple, le package pynsee propose des options qui seront retranscrites automatiquement dans l'URL de requête pour faciliter le travail sur les données Insee.

Voyons un exemple de code :

```
import requests
import pandas as pd

url_api ="https://fr.openfoodfacts.org/cgi/search.pl?action=process&
    tagtype_0=categories&tag_contains_0=contains&tag_0=biscuits&fields=
    generic_name&page_size=100&json=true"

req = requests.get(url_api)

wb = req.json()
df = pd.json_normalize(wb["products"])
print(df)
```

Ce code contacte la base de données française du site OpenFoodFacts et récupèrent les noms des cent premiers produits de la catégorie biscuits . Regardons en détail comment il fonctionne.

On commence par importer deux bibliothèques, pandas, qui permettra de travailler sur les données une fois extraites et requests qui va permettre d'effectuer les requêtes à l'API.

On construit ensuite l'url complète de la requête. Elle est composée de l'url de base de l'API `"https://fr.openfoodfacts.org/cgi/search.pl?action=process"`, puis d'un ensemble de paramètres `"&tagtype_0=categories&tag_contains_0=contains&tag_0=biscuits&fields=generic_name&page_size=100&json=true"`.

Notez qu'en général, on ne colle pas l'url comme ça d'un coup, mais on définit les divers paramètres dans des sous-chaines, avant de les fusionner, pour faciliter le reparamétrage des requêtes.

On lance ensuite la requête, dont le résultat est stocké dans `req`. L'étape suivante est d'extraire le json de la requête, puis de transformer le json en un dataframe, qui sera plus facile à traiter.

Le travail principal lorsqu'on code avec une API est de bien comprendre comment elle fonctionne, et cela à deux niveaux :

1. Au niveau de la construction de la requête, pour savoir quels sont les paramètres et leur format.
2. Au niveau du format des résultats, pour bien comprendre les différents champs de données fournies par l'API

Je vous rappelle que vous pouvez trouver la documentation de la bibliothèque pandas à cette adresse : https://pandas.pydata.org/docs/index.html

Zoologies des API [Cours]

Attention, toutes les API ne sont pas nées égales. La plupart des API mettent en place des limitations pour alléger leurs serveurs . On pourra citer :

- L'authentification. Certaines API ne sont disponibles qu'aux utilisateurs identifiés. Après s'être inscrit sur le site de l'API, des identifiants vous seront fournis qu'il faudra intégrer à la requête.
- La limitation en volumes. De nombreuses API limitent le nombre de requêtes faisable d'une même adresse ip . On est souvent sur un maximum de requête par heure.
- Les API payantes ou freemium.

Vous pourrez par ailleurs trouver plein d'annuaires d'API en ligne, par exemple : https://rapidapi.com/categories ou https://api.gouv.fr/rechercher-api

Exercice 1 (,)

On va se servir de l'API du site Open Food Facts, car elle ne nécessite pas d'authentification et ne pose aucune contrainte sur le nombre de requêtes. Vous pourrez trouver la documentation de l'API ici : https://openfoodfacts.github.io/openfoodfacts-server/api/ .

1. *Commençons par étudier le format des requêtes. Effectuez la requête permettant d'afficher le produit de code barre 7613032413491 avec l'url suivante* `https://fr.openfoodfacts.org/api/v0/product/7613032413491.json`. *Comparez le contenu de cette requête avec la page web du produit https://fr.openfoodfacts.org/produit/7613032413491/crunch-snack-nestle pour repérer les champs importants (le nom du produit, sa marque, à quelles catégories il appartient, ses ingrédients, ses valeurs nutritionnelles, son nutriscore/ecoscore,...)*

2. *Quels sont les produits correspondant aux code-barres suivant ? 5449000131805, 3155250358788, 5411188119098, 8715700407760*
3. *Combien y-a-t'il de produits de marque heudebert ? Récupérez leurs noms.*
4. *Récupérez les noms et nutriscore de tous les sodas. Quel est le produit avec le meilleur nutriscore ?*

□

Exercice 2 (,)

En utilisant l'API gratuite d'open météo https://open-meteo.com/, vous produirez une visualisation comparant la météo Aurillacoise sur les années 2023, 2013, 2003 ,1993 et 1983. □

Exercice 3 (Projet,)

En utilisant l'API du Metropolitan Museum of New York https://metmuseum.github.io/ et d'éventuelles données supplémentaires, vous produirez un tableau de bord offrant un état des lieux des collections du musée suivant un axe d'étude de votre choix. Vous devrez rendre tous les codes et les fichiers de données intermédiaires. □