TP Guidé : WebScrapping

S. Gibaud

10 novembre 2021

I. Le WebScrapping dans la vraie vie

Le WebScrapping est une activité très utilisé dans l'industrie informatique. En effet c'est un outil très efficace pour récolter des données sur Internet et pouvoir les traiter. Cependant est ce que c'est légal ?

Extrait de *L'usine Digitale*

La pratique du web scraping pourrait être considérée comme un "vol de données" (atteinte au STAD) en s'appuyant sur l'article 323-3 du Code pénal qui énonce :

"Le fait d'introduire frauduleusement des données dans un système de traitement automatisé, d'extraire, de détenir, de reproduire, de transmettre, de supprimer ou de modifier frauduleusement les données qu'il contient est puni de cinq ans d'emprisonnement et de 150 000 € d'amende. Lorsque cette infraction a été commise à l'encontre d'un système de traitement automatisé de données à caractère personnel mis en oeuvre par l'Etat, la peine est portée à sept ans d'emprisonnement et à 300 000 € d'amende." Il conviendrait toutefois de caractériser l'intention frauduleuse du web scraping.

On peut également considérer qu'il s'agit d'un acte de concurrence déloyale ou d'une pratique de parasitisme, la personne recourant au web scraping n'ayant pas effectué les mêmes efforts (en termes de communication, d'investissements techniques, etc.) que le teneur du site pour collecter ces données.

En outre, ce point se double d'exigences impératives en matière de protection des données à caractère personnel des utilisateurs à l'heure où le RGPD constitue une donne incontournable dans la stratégie des entreprises connectées. Il met en exergue la nécessité d'une gouvernance des données à caractère personnel (tant organisationnelle que juridique ou technique)...

Le mot du professeur

Pour ne pas tomber dans l'illégalité dans le webscrapping.

— Ne publiez pas en ligne les informations que vous avez récupérer avec le webscrapping.
— Limitez au maximum le nombre de connexion au site (1 connexion maximum par page), comme vous le feriez en naviguant à la main. Les connexions sont faites grâce au module *requests*. **NE METTEZ JAMAIS *requests* DANS UNE BOUCLE**

Si les données sont en licence libre, alors vous avez le droit d'y accéder librement.

II. Installation des prerequis

Taper dans la console :

```
pip install beautifulsoup4
```

Cette commande permet d'installer la librairie beautifulsoup4 qui permet à partir du texte d'une page html d'obtenir un objet dans lequel on pourra naviguer.

III. Initialisation

On va charger les librairies dont on aura besoin :

```
import requests
import urllib.request
import time
from bs4 import BeautifulSoup
import os
```

— Les ligne 1 et 2 importe le module requests qui permet de faire des requêtes à des server web.
— La ligne 3 permet d'importer le module time qui permettra au programme d'attendre (que la page se charge par exemple)
— La ligne 4 importe le module beautifulsoup qui permet de rendre les pages html "intelligible"
— la ligne 5 importe le package os qui permet de manipuler l'os.

Entrer dans une variable url l'url du site qui vous intéresse. Ici dans ce tp pour vous guider on va prendre : "https ://www.lnr.fr/rugby-top-14/classement-rugby-top-14".

IV. Navigation sur le site

1. Navigation à la main

On va chercher où est la meilleure équipe de tout les temps dans le code html. Pour cela, entrer dans le navigateur l'url et faîte un clic droit sur **le stade Toulousain**.

▪ Exercice .1.

1. Donner la classe ("class") de l'objet que vous avez sélectionné.
2. Vérifier que la classe du parent de l'objet que vous avez sélectionné est "row-n-1".
3. Vérifier que le type du grand-père de l'objet que vous avez sélectionné est "table" et que ses classes sont "ranking-dividers" et "sorted-by-rank-asc" (ce dernier peut être modifié si vous avez touché au tri du tableau)

▪

2. Navigation du site en Python

▪ Exercice .2.

1. Dans l'éditeur entrer :

```
response = requests.get(url)
soup = BeautifulSoup(response.text, "html.parser")
```

   En utilisant votre cerveau et la section III. que font ses deux lignes. Vous pouvez voir *soup* dans la console pour avoir une idée de ce que vous obtenez.
2. Le nom de l'équipe étant dans un tableau "Td", nous allons récupérer toutes les balises td du fichier html. Pour cela affecter à une variable TD *soup.findAll("td")*. Quelle est le type de TD ? quelle est la longueur de TD ?
3. chaque élément de TD fonctionne comme un dictionnaire avec plusieurs clé. Une de ces clé est "class". Faites une boucle qui affiche la "class" de tous les éléments de TD.
4. Le nom de l'équipe est dans les balises avec la class 'views-field-field–quipe'. Faite une boucle sur TD qui affiche *True* si l'élément de la liste a 'views-field-field–quipe' dans ses "class" et *False* sinon. On pourra utiliser : *'views-field-field–quipe' in td["class"]* avec *td* l'élement de TD.
5. Pour obtenir le nom du club à partir d'un élément *td* de *TD* on fait : *td.findChild().text*. Faîtes une boucle qui affiche le nom de toutes les équipes. En petit groupe proposer une explication de l'instruction *td.findChild().text*.
6. Faites une boucle qui affiche *True* si l'élément *td* de *TD* représente le Stade Toulousain et *False* sinon. Vous devez avoir exactement 1 seul *True*. (Les espaces peuvent vous jouer des tours, vous pouvez utiliser *in*).
7. Pour obtenir toutes les informations sur les résultats du stade toulousain on doit aller voir sa balise parente. Pour se faire on va affecter à *Parent td.findParents()[0]*. Les données seront dans les enfants du parent (il faudra utiliser *.findChildren()*). Donner le nombre d'enfants de *Parent*.
8. Explorer dans la console à la main les enfants de *Parent*. Donner les index des enfants donnant : le rang (ranking), les points (field-point), la journée, le nombre de victoires, de défaites, de nuls, de bonus, de points marqués, de points concédés, de différences de points.
9. On peut obtenir le nombre dans l'enfant en ajoutant après l'enfant un *.text*. Par exemple

```
>>>Children
   [15].text


     +109
```

Faire une fonction qui prend en argument une liste d'enfants et qui affiche les statistiques de la question 8.

10. Faire une boucle qui affiche les statistiques de toutes les équipes (penser à mettre le nom des équipes.)
11. Faire une fonction qui encapsule qui affiche les statistiques de toutes les équipes.

V. Devoir Maison

Sur le site https ://fbref.com/fr/pays/joueurs/FRA/Joueurs-de-football-de-France, combien il y a-t-il de joueurs de foot ayant un prénom qui commence par un "A" et qui ont "MF" à la suite de leur nom. Donner le code en python qui vous a permis de trouver la réponse.