



NBA Players Scrapping :

ROUICHI ADIL 17806389

adil.rouichi011@gmail.com



Descriptif projet :

Scraper en python des statistiques et informations principales des tous les joueurs de NBA dans le but de créer un dataset et d'effectuer des opérations sur celui-ci.

Le programme permet de :

- ➔ Récouter un dataset de joueurs selon certains paramètres.
- ➔ Créer une ou plusieurs collection(s) à partir de ce dataset selon des suites de paramètres. (Ces collections peuvent être exportées en CSV).
- ➔ Créer une ou plusieurs fiche(s) de statistique(s) entre ces collections selon un attribut de joueur en particulier. (Peut aussi être exporté en CSV).

Informations :

Le site source pour le scraping est <https://www.nba.com/>.

La liste de joueurs dont dispose le site comporte les joueurs actuels et ainsi que l'historique des joueurs ayant joués en NBA.

Le nombre de joueurs maximum est 4731.

Ce scraper constitue son dataset en deux étapes :

- ➔ Il recueille l'ID de tous les joueurs selon quelques attributs grâce à une requête http GET envoyée à l'API du site web via la librairie **Requests**. Ces ID sont stockés dans un vecteur et concaténés de la manière suivante : « <https://www.nba.com/stats/player> » + ID + « /career ».
- ➔ Une boucle parcourt toutes les URL de joueurs stockées dans la liste et extrait les données de chaque joueur afin de créer le dataset qui est une liste de dictionnaires comportant les informations de chaque joueur.

```
Dictionnaire_joueur = {  
    'Id' : '',  
    'Name' : '',  
    'Surname' : '',  
    'Number' : '0',  
    'Position' : '',  
    'Ppg' : '0',  
    'Rpg' : '0',  
    'Apg' : '0',  
    'Height' : '0',  
    'Weight' : '0',  
    'Country' : '',  
    'College' : '',  
    'Birthdate' : '00/00/0000',  
    'Experience' : '0'  
}
```

Les données de chaque joueur sont extraites grâce à une requête HTML sur l'URL du joueur.

Ensuite elles sont découpées grâce à la librairie **Beautiful Soup** qui me permet de parser ses données. Je retrouve les informations qui m'intéressent en m'aidant des classes ou id des tags HTML et en utilisant des regex afin de récupérer les données qui m'intéressent.

Nécessite les librairies « requests » et « BeautifulSoup ».

Utilisation du programme :

Personnalisation de la requête :

Cette partie permet de filtrer directement selon certains critères dès la requête.

Liste des arguments :

- "country=value|" value == un des pays appartenant à la variable parameter_country_list du fichier python. (Filtre tous les joueurs de cette nationalité)
- "college=value|" value == un des pays appartenant à la variable parameter_college_list du fichier python. (Filtre tous les joueurs de cette université)
- "draftyear=value|" value == une année correspondant à la date d'entrée en NBA du joueur (compris entre 1947 et 2021)
- "historical=value|" value == 0 : les joueurs encore en compétition | value == 1 : tous les joueurs ayant joués et jouant encore
- "limit=value|" value == int : limite le nombre de joueurs scraper à value

Création des collections :

Cette partie permet en combinant certains paramètres de faire une sélection dans le dataset afin de créer une nouvelle collection. Dans cette partie vous pouvez combiner autant de arguments que vous voulez dont des paramètres de même type.

Exemple : Ppg>10 | Ppg<15 | Country!=USA | Birthdate>01/11/1998| (pensez à terminer par un "|" (pipe)).

Liste des arguments : (cette fois-ci c'est un "==" et non un "=" pour l'égalité)

OPERATEURS DISPONIBLES : "=", "!=", "<", ">"

- "Number==value|" (numéro du joueur) value == int
- "Ppg==value|" (points par match) value == int
- "Rpg==value|" (rebonds par match) value == int
- "Apg==value|" (passes décisives par match) value == int
- "Height==value|" (taille du joueur) value sous la forme 0.00 (exemple : 1.98 ; 1.86 ; 2.05 ; 2.1)
- "Weight==value|" (poids du joueur) value == int
- "Birthdate==value|" (date de naissance du joueur) value sous la forme dd/mm/yyyy (exemple : 01/11/1998)

OPERATEURS DISPONIBLES : "==" , "!="

- "Name==value|" (prénom du joueur) value == word
- "Surname==value|" (nom du joueur) value == word
- "Position==value|" (poste du joueur) value == Guard or Center or Forward
- "Country==value|" (nationalité du joueur) value == un des pays appartenant à la variable parameter_country_list du fichier python.
- "College==value|" value == un des pays appartenant à la variable parameter_college_list du fichier python.

LES PARAMETRES DE LA REQUETE DE SELECTION FONCTIONNENT COMME S'ILS ETAIENT LIES PAR UN ET LOGIQUE :

➔ Height>2.1 | Weight>80 : mesure plus de 2m10 et pèse plus de 80kg

Création des fiches statistiques :

Vous pouvez créer des feuilles de statistiques par rapport aux diverses collections précédemment créer. Par exemple, vous pouvez afficher les statistiques sur les points par matchs, les années d'expérience, les dates de naissances, la répartition en fonction des pays, universités ou positions de jeu.

Liste des attributs :

➔ Ppg, Rpg, Apg, Height, Weight, Birthdate, Experience, Position, Country, College.

----> A écrire comme ci-dessus sans caractères supplémentaires. Aucune combinaison n'est possible, les statistiques ne sont disponibles que pour un attribut à la fois.

Export de collections ou statistiques CSV :

- ➔ Vous pouvez choisir d'exporter les collections précédemment créer en fichier CSV dans le dossier "collection".
- ➔ Vous pouvez ensuite choisir d'exporter les feuilles de statistiques précédemment créer en fichier CSV dans le dossier "stats".

**Merci pour votre attention, en espérant que mon programme vous plaît.
Bonne continuation.**