



On veut utiliser l'api de la nba pour afficher des informations sur les joueurs , les équipes de la nba.

la documentation peut être obtenue sur la page github : https://github.com/swar/nba_api.git

Pour se connecter à l'api , on commence par importer les modules:

```
1 from nba_api.stats.static import players
2 from nba_api.stats.endpoints import playercareerstats
3 import pandas as pd
```

On importe le module pandas car les données peuvent souvent être exploitées en passant par des dataframes.

La bibliothèque `players` et sa méthode `get_players` permettent d'obtenir une liste de dictionnaires

```
1 #une liste de dictionnaires
2 #chaque dictionnaire contient les infos sur un joueur
3 tous_les_joueurs = players.get_players()
4 #print(tous_les_joueurs)
```

Q1 Quelles sont les clés de ces dictionnaires?

Q2 Ecrire une instruction pour récupérer les dictionnaires où le nom du joueur est 'Victor'.

On peut aussi récupérer les infos sur un joueur directement avec la méthode `find_players_by_first_name`

```
1 #print(players.find_players_by_first_name('LeBron'))
2
3 #ou alors avec le nom complet
4 #méthode find_players_by_first_name
5 print(players.find_players_by_full_name('Rudy Gobert'))
```

Q3 Chercher l'ID du joueur à partir de son nom et prénom.

Q4 Ecrire une instruction pour récupérer la liste de tous les noms et prénoms de tous les joueurs

Q5 Ecrire une instruction pour récupérer la liste de tous les joueurs qui s'appellent Williams

Pour récupérer les statistiques d'un joueur , on utilise l'endpoint `playercareerstats`

```
1
2 #on récupère l'identifiant de LeBron James
3 id_joueur = all_players[all_players.full_name=='LeBron James']['id']
4 #print(id_joueur)
5
6 # Appeler l'endpoint pour les statistiques de carrière
7 career = playercareerstats.PlayerCareerStats(player_id=id_joueur)
8
9 # Transformer les données en DataFrame Pandas
10 df = career.get_data_frames()[0]
```

Q6

Quelles sont les identifiants de champs ?

Ecrire une instruction pour les afficher.

On peut en sélectionner quelques colonnes:

```
1 extrait = ['SEASON_ID', 'TEAM_ABBREVIATION']  
2 print(df[extrait])
```

Q7 Ecrire une instruction qui affiche les statistiques sur le joueur Victor Wembanyama.

Q8 Ecrire une fonction `statistiques_joueur(nom_et_prenom_du_joueur)` qui prend pour paramètre le nom et prénom d'un joueur et qui affiche les statistiques sur ce joueur.

1. créer l'arborescence suivante:

```
$ mkdir projet_nba
Dans ce répertoire créer un fichier nommé app.py
$mkdir templates
puis créer un fichier index.html
```

2. Dans le fichier app.py copier le code suivant

```
1  from flask import Flask, render_template
2  from nba_api.stats.endpoints import leaguestandingsv3
3
4  app = Flask(__name__)
5
6  @app.route('/')
7  def home():
8  try:
9  # 1. Appel à l'API NBA pour obtenir le classement actuel
10 standings = leaguestandingsv3.LeagueStandingsV3()
11
12 # 2. Conversion des données en dictionnaire Python (JSON)
13 data = standings.get_dict()
14
15 # On récupère la liste des équipes et leurs stats
16 # La structure de nba_api est souvent : data['resultSets'][0]['rowSet']
17 headers = data['resultSets'][0]['headers']
18 rows = data['resultSets'][0]['rowSet']
19
20 # 3. On envoie les données au fichier HTML
21 return render_template('index.html', headers=headers, teams=rows)
22
23 except Exception as e:
24 return f"Erreur lors de la récupération des données : {e}"
25
26 if __name__ == '__main__':
27 app.run(debug=True)
```

3. puis copier dans le fichier index.html le code suivant

```
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4 <meta charset="UTF-8">
5 <title>Stats NBA Live</title>
6 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/
  bootstrap@5.3.0/dist/css/bootstrap.min.css">
7 </head>
8 <body class="container mt-5">
9 <h1 class="mb-4">Classement NBA</h1>
10
11 <table class="table table-striped">
12 <thead class="table-dark">
13 <tr>
14 <th>Equipe</th>
15 <th>Victoires</th>
16 <th>Defaites</th>
17 <th>% Victoires</th>
18 </tr>
19 </thead>
```

```
20 <tbody>
21 {% for team in teams %}
22 <tr>
23 <td>{{ team[5] }} {{ team[6] }}</td> <td>{{ team[13] }}</td>
    <td>{{ team[14] }}</td> <td>{{ team[15] }}</td>
    </tr>
24 {% endfor %}
25 </tbody>
26 </table>
27 </body>
28 </html>
```

4. exécutez le fichier app.py puis dans votre navigateur préféré tapez:
<http://127.0.0.1:5000>
5. Adaptez ce qui précède pour afficher des statistiques sur le joueur LeBron James