User Documentation

Gabriel Garces Chavez

Doujana Alhabib

Emmanuel Olaoluwa ADENIYI

NBA Players Information Scraping

A python application that python, flask to scrape from NBA players database reference.
This project provides a page in which you can serve all the user needs from add, delete, search for a player and we will also provide the code for everything we made to create the functionality of the code. The data-set contains aggregate individual statistics for all NBA seasons from the beginning of the NBA, but only the players who start their career from 1950 are considered. from basic box-score attributes such as points, assists, rebounds etc., to more advanced money-ball like features such as Value Over Replacement

SCRAPE STRATEGY

myScrape.py

Var URLBASE - This is the usr base for referencing players of different names. All arranged alphabetically.

Def get_player_image() -
Keyword arguments
player_detail_link - url for the player profile

- Open player link
- Call beautiful soup function passing and parsing with the default html-parser
- Find tags and meta of player in DOM where id = ‘info’ and id =’meta’
- Check if the player has an image. If yes assign image src else img is assigned not found return image

Def Scrape_player_Data(list_letters):
Parameter : list _letters - the list letters is from a - z which will be added to the base url to retrieve the page of the players name. Contains first letter of name to search for
Table_body - Since each page contains a body we find the tag ‘tbody’

Players row = for every tr there are 10 table data. For every row get ‘table data’, get the year of career which must be greater or equal to 1950.

Row.th.a.text is the first table data of the row containing the first and last name , the height, weight, birthday, college and detail link are looped over and appended to a list player_info.

DEF search_playerby_name(name_search):

Search player by their first name

```
Keyword arguments
   list_letters -- contains first letter of name to look for
```

It searches through the pages for players a-z for the name. It can be slow since it opens each player's page to find each link so it has a limit of 50 items.

Pattern for searching for name: name like name_search to lower case to ensure consistency

For each letter in the alphabet:

Url_base + letter

Table_body = First we look for the body of the DOM, then narrow down looking for the table row

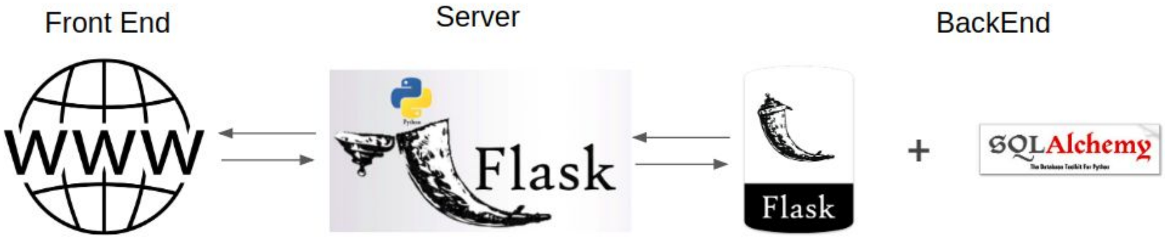
In each row found find the data of each row ‘td’

Players row = for every tr there are 10 table data. For every row get ‘table data’, get the year of career which must be greater or equal to 1950.

Row.th.a.text is the first table data of the row containing the first and last name , the height, weight, birthday, college and detail link are looped over and appended to a list player_info.

Further into detail_link - url_base + row . tablehead . link ‘a’. Append scrapped date to array player_info

ARCHITECTURE:



architecture of the web application:

In here we can see the web application architecture which is a pattern of interaction between various web application components. The picture is showing how the web application architecture depends on how the application logic is distributed among the client and server sides.

Why we are using web scraping:

extract and process large amounts of data from the web. analyzes large amounts of datasets, the ability to scrape data from the web is a useful skill to have.and web scraping using Python is a skill you can use to extract the data into a useful form that can be imported.

Requirement to run program:

Download the project and run pip install -r requirements.txt After that run python nbaWeb.py -e dev to start server. Or use python nbaWeb.py -h to show basic commands for scraping without starting the server.

References:

http://www.basketball-reference.com/