

ESPN Cricket Data Scraping

Requirements:-

```
BeautifulSoup -
requests -
sqlite3 - A light-weight database to store the information fetched for further querying and
analysis, If automated Jog to be done by using like Flask (Also converted to CSV files)
Others - argparse, logging
```

Program Structure:-

```
|-- CSV_Converted_Data
|   |-- Batting_Stats_Odi.csv
|   |-- Batting_Stats_T20.csv
|   |-- Bowling_Stats_Odi.csv
|   |-- Bowling_Stats_T20.csv
|   |-- Countries.csv
|   |-- Players.csv
|-- DEFAULT_DB.sqlite
|-- ESPN_Cricket_Data_Scraper.py
|-- ESPN_Cricket_Data_Scraping.md
|-- StoredDB.sqlite
|-- espn_cricket_data_scraper.log
```

About:-

This program provides user the flexibility to choose the countries of whose player statistics are to be scraped and stored in database along with either choosing ODI/T20 or both.

To Execute:- Now for terminal (command line)

```
>>> ESPN_Cricket_Data_Scraper.py [-h] [-d DATABASE_NAME] [-t TYPE_OF_MATCH] [-c [COUNTRIES
[COUNTRIES ...]]]
```

This with Default values:- Scrapes all players statistics(bowling and fielding) from major 11 playing nations in both ODI and T20 format.

```
example: python ESPN_Cricket_Data_Scraper.py
-d : 'DEFAULT_DB'
-t : 'ALL'
-c : 'ALL'
```

Usage with custom entries:-

```

Example:- (Run)
>>> python ESPN_Cricket_Data_Scraper.py -d StoredDB -t ODI -c [india, pakistan]

-d :
-t : ODI/T20/ALL
-c [australia,bangladesh,england,india,new-zealand,pakistan,south-africa,sri-lanka,west-indies,zimbabwe,afghanistan] ##select based on requirement in list

```

The execution of above command would scrape the website for batting/bowling stats in ODI of all players from india and pakistan and parse and store results in 'StoredDB.sqlite' **database** which could be queried using below table structure.

The **sqlite database** would have the below table structure where the scraped data is parsed and stored in a normalized way:

```

Countries > country_id PRIMARY KEY,country

Players > country_id ,player_id UNIQUE,player ,odi_cap ,t20_cap

Batting_Stats_Odi > player ,playing_span ,matches_played ,innings_batted ,not_outs ,
runs_scored ,highest_innings_score ,batting_average ,balls_faced ,batting_strike_rate
,hundreds_scored ,scores_between_50_and_99 ,ducks_scored ,boundary_fours ,boundary_sixes

Bowling_Stats_Odi > player ,playing_span ,matches_played ,innings_bowled_in ,overs_bowled
,balls_bowled ,runs_conceded ,maidens_earned ,wickets_taken ,best_bowling_in_an_innings
,bowling_average ,economy_rate ,bowling_strike_rate ,four_wkts_exactly_in_an_inns
,five_wickets_in_an_inns

Batting_Stats_T20 > player ,playing_span ,matches_played ,innings_batted ,not_outs ,
runs_scored ,highest_innings_score ,batting_average ,balls_faced ,batting_strike_rate
,hundreds_scored ,scores_between_50_and_99 ,ducks_scored ,boundary_fours ,boundary_sixes

Bowling_Stats_T20 > player ,playing_span ,matches_played ,innings_bowled_in ,overs_bowled
,balls_bowled ,runs_conceded ,maidens_earned ,wickets_taken ,best_bowling_in_an_innings
,bowling_average ,economy_rate ,bowling_strike_rate ,four_wkts_exactly_in_an_inns
,five_wickets_in_an_inns

```

Once the parser completes execution we can query the required information for further **analysis** referring the table structure provided above and also can be converted into csv file for data analysis for pandas point of view.

Some Snap-shoots of scraped Cricket data:-

1. Batting stats of ODI:-

DB Browser for SQLite - /home/vijendra/Desktop/India_in_Pixels/python-web-scaping-cricket-stats/testDB.sqlite

File Edit View Help

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragma Execute SQL

Table: Batting Stats_Odi

	player	playing_span	natches_player	innings_batted	not_outs	runs_scored	nest_innings
1	G Boycott	1971-1981	36	34	4	1082	105
2	MC Cowdrey	NA	1	1	0	1	1
3	BL D'Oliveira	1971-1972	4	4	1	30	17
4	JH Edrich	1971-1975	7	6	0	223	90
5	KWR Fletcher	1971-1982	24	22	3	757	131
6	JH Hampshire	1971-1972	3	3	1	48	25*
7	R Illingworth	1971-1973	3	2	0	5	4
8	APE Knott	1971-1977	20	14	4	200	50
9	P Lever	1971-1975	10	3	2	17	8*
10	K Shuttlewo...	NA	1	1	0	7	7
11	JA Snow	1971-1975	9	4	2	9	5*
12	DL Amiss	1972-1977	18	18	0	859	137
13	GG Arnold	1972-1975	14	6	3	48	18*
14	DB Close	NA	3	3	0	49	43
15	AW Greig	1972-1977	22	19	3	269	48
16	RA Woolmer	1972-1976	6	4	0	21	9

1 - 17 of 558

Go to: 1

Edit Database Cell

Mode: Text

Import Export Set as NULL

Type of data currently in cell: NULL

0 byte(s)

SQL Log

Show SQL submitted by Application

Clear

```

1 PRAGMA foreign_keys = "1";
2 SELECT type,name,sql,tbl_name,'0' AS temp FROM sqlite_master UNI
3 PRAGMA encoding
4 SELECT COUNT(*) FROM (SELECT `rowid`,`*` FROM `Batting Stats_Odi
5 SELECT `rowid`,`*` FROM `Batting Stats_Odi` ORDER BY `rowid` A
6 SELECT COUNT(*) FROM (SELECT `rowid`,`*` FROM `Batting Stats_Odi
7 SELECT `rowid`,`*` FROM `Batting Stats_Odi` ORDER BY `rowid` A
8

```

SQL Log Plot DB Schema Remote

UTF-8

2. Countries Name:-

DB Browser for SQLite - /home/vijendra/Desktop/India_in_Pixels/python-web-scaping-cricket-stats/testDB.sqlite

File Edit View Help

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragma Execute SQL

Table: Countries

	country_id	country
1	1	england
2	2	australia
3	3	south-africa
4	4	west-indies
5	5	new-zealand
6	6	india
7	7	pakistan
8	8	sri-lanka
9	9	zimbabwe
10	25	bangladesh
11	40	afghanistan

1 - 11 of 11

Go to: 1

Edit Database Cell

Mode: Text

Import Export Set as NULL

Type of data currently in cell: NULL

0 byte(s)

SQL Log

Show SQL submitted by Application

Clear

```

2 SELECT type,name,sql,tbl_name,'0' AS temp FROM sqlite_master
3 PRAGMA encoding
4 SELECT COUNT(*) FROM (SELECT `rowid`,`*` FROM `Batting Stats
5 SELECT `rowid`,`*` FROM `Batting Stats_Odi` ORDER BY `rowid`
6 SELECT COUNT(*) FROM (SELECT `rowid`,`*` FROM `Batting Stats
7 SELECT `rowid`,`*` FROM `Batting Stats_Odi` ORDER BY `rowid`
8 SELECT COUNT(*) FROM (SELECT `rowid`,`*` FROM `Countries` OR
9 SELECT `rowid`,`*` FROM `Countries` ORDER BY `rowid` ASC LI
10 SELECT COUNT(*) FROM (SELECT `rowid`,`*` FROM `Countries` OR
11 SELECT `rowid`,`*` FROM `Countries` ORDER BY `rowid` ASC LI
12

```

SQL Log Plot DB Schema Remote

UTF-8

3. Players Name:-

DB Browser for SQLite - /home/vijendra/Desktop/India_In_Pixels/python-web-scaping-cricket-stats/testDB.sqlite

File Edit View Help

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragma Execute SQL

Table: Players

	country_id	player_id	player	odi_cap	t20_cap
245	1	414990	JT Ball	Y	Y
246	1	521637	BM Duckett	Y	Y
247	1	370535	TS Roland-J...	Y	NULL
248	1	550235	TK Curran	Y	Y
249	1	464626	C Overton	Y	NULL
250	1	662973	SM Curran	Y	NULL
251	1	457279	OP Stone	Y	NULL
252	1	669855	JC Archer	Y	Y
253	1	364788	BT Foakes	Y	Y
254	1	236489	DJ Malan	Y	Y
255	1	20073	CP Schofield	NULL	Y
256	1	23911	A Khan	NULL	Y
257	1	459257	TS Mills	NULL	Y
258	1	660889	MS Crane	NULL	Y
259	1	403902	LS Livingsto...	NULL	Y
260	2	4558	GS Chappell	Y	NULL
261	2	4560	IM Channell	Y	NULL

245 - 261 of 2053

Go to: 1

Edit Database Cell

Mode: Text

Import Export Set as NULL

Type of data currently in cell: NULL

0 byte(s)

SQL Log

Show SQL submitted by: Application

Clear

```

44 SELECT COUNT(*) FROM (SELECT `rowid`,`*` FROM `Bowling Stats`
45 SELECT `rowid`,`*` FROM `Bowling Stats Odi` ORDER BY `wickets`
46 SELECT COUNT(*) FROM (SELECT `rowid`,`*` FROM `Countries` ORD
47 SELECT `rowid`,`*` FROM `Countries` ORDER BY `rowid` ASC LI
48 SELECT COUNT(*) FROM (SELECT `rowid`,`*` FROM `Countries` ORD
49 SELECT `rowid`,`*` FROM `Countries` ORDER BY `rowid` ASC LI
50 SELECT COUNT(*) FROM (SELECT `rowid`,`*` FROM `Players` ORDER
51 SELECT `rowid`,`*` FROM `Players` ORDER BY `rowid` ASC LIM
52 SELECT COUNT(*) FROM (SELECT `rowid`,`*` FROM `Players` ORDER
53 SELECT `rowid`,`*` FROM `Players` ORDER BY `rowid` ASC LIM
54

```

SQL Log Plot DB Schema Remote

UTF-8

Querying From DataBase:-

DB Browser for SQLite - /home/vijendra/Desktop/India_In_Pixels/python-web-scaping-cricket-stats/testDB.sqlite

File Edit View Help

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragma Execute SQL

SQL 1 ✖

```

1 SELECT * FROM Players;

```

	country_id	player_id	player	odi_cap	t20_cap
1	1	9187	G Boycott	Y	NULL
2	1	10846	MC Cowdrey	Y	NULL
3	1	11914	BL D'Oliveira	Y	NULL
4	1	12490	JH Edrich	Y	NULL
5	1	12854	KWR Fletcher	Y	NULL
6	1	14024	JH Hampshire	Y	NULL

2053 rows returned in 11ms from: SELECT * FROM Players;

Edit Database Cell

Mode: Text

Import Export Set as NULL

Type of data currently in cell: NULL

0 byte(s)

SQL Log

Show SQL submitted by: Application

Clear

```

9 SELECT `rowid`,`*` FROM `Countries` ORDER BY `rowid` ASC LI
10 SELECT COUNT(*) FROM (SELECT `rowid`,`*` FROM `Countries` ORD
11 SELECT `rowid`,`*` FROM `Countries` ORDER BY `rowid` ASC LI
12 SELECT COUNT(*) FROM (SELECT `rowid`,`*` FROM `Players` ORDER
13 SELECT `rowid`,`*` FROM `Players` ORDER BY `rowid` ASC LIM
14 SELECT COUNT(*) FROM (SELECT `rowid`,`*` FROM `Players` ORDER
15 SELECT `rowid`,`*` FROM `Players` ORDER BY `rowid` ASC LIM
16 SELECT COUNT(*) FROM (SELECT * FROM `Players`);
17 SELECT * FROM `Players` LIMIT 0, 50000;
18 SELECT type,name,sql,tbl_name,'0' AS temp FROM sqlite_master
19

```

SQL Log Plot DB Schema Remote

UTF-8

The screenshot shows the DB Browser for SQLite interface. The main window displays the results of a query: `SELECT * FROM Batting_Stats_Odi;`. The results are shown in a table with 8 columns: player, playing_span, matches_played, innings_batted, not_outs, runs_scored, and highest. The table contains 558 rows of data. The right sidebar shows the 'Edit Database Cell' panel with 'Mode: Text' and 'Type of data currently in cell: NULL'. The 'SQL Log' panel shows the submitted query.

	player	playing_span	matches_played	innings_batted	not_outs	runs_scored	highest
1	G Boycott	1971-1981	36	34	4	1082	105
2	MC Cowdrey	NA	1	1	0	1	1
3	BL D'Oliveira	1971-1972	4	4	1	30	17
4	JH Edrich	1971-1975	7	6	0	223	90
5	KWR Fletcher	1971-1982	24	22	3	757	131
6	JH Hampshire	1971-1972	3	3	1	48	25*
7	R Illingworth	1971-1973	3	2	0	5	4

Due to time constraint I couldn't join all three tables Batting_Stats_Odi, Countries and Players, but I definitely will do it. Following will be final result of all three column of csv file whci can be easily **Export**.

The screenshot shows the DB Browser for SQLite interface. The main window displays the results of a complex SQL query. The query is: `SELECT Players.player AS player, Players.country_id AS play_country_id, Countries.country AS country, Countries.country_id AS country_cid, Batting_Stats_Odi.player AS BSO_player FROM Players LEFT JOIN Countries ON play_country_id = country_cid LEFT JOIN Batting_Stats_Odi ON player = BSO_player WHERE Batting_Stats_Odi.playing_span BETWEEN 1970 AND 2019;`. The results are shown in a table with 4 columns: player, runs_scored, country, and country_cid. The table contains 4 rows of data. The right sidebar shows the 'Edit Database Cell' panel with 'Mode: Text' and 'Type of data currently in cell: Text / Numeric'. The 'SQL Log' panel shows the submitted query.

	player	runs_scored	country	country_cid
1	G Boycott	1082		
2	BL D'Oliveira	30		
3	JH Edrich	223		
4	KWR Fletcher	757		

Note:- There are a lot to do , but for now it's done. (Every details like maximum run, Maximum wickets, Hundreds, Fifties, Sixes, Fours etc.)

Here I didn't scrape IPL data which is an easy task.

This is my approach for data scraping it's interesting, but data analysis will be much more interesting. I hope, I will get the chance to work for such start-up and level up my skills.