

Tutorial API SportsDataIO

Nome: *Giovanni*

Cognome: *Imbesi*

Matricola: *1000006253*

Corso: Social Media Management

Docente: Antonino Furnari

Anno Accademico: *2021/2022*

Introduzione

Fondata nel 2008 come FantasyData LLC, SportsDataIO è un fornitore di dati sportivi che copre decine di migliaia di eventi sportivi durante tutto l'anno. Nel corso del primo decennio, hanno ampliato la copertura per tutti i principali campionati di calcio nordamericani e internazionali, con un'offerta di contenuti che fornisce dati, notizie e immagini a una vasta gamma di settori, tra cui giochi virtuali, editoria, trasmissione, campionati e squadre. Più recentemente, con l'abrogazione di PASPA, hanno cambiato nome in SportsDataIO e replicato la precedente missione di fornire feed virtuali specifici all'industria dei giochi virtuali insieme a una suite di widget per completare le API come metodo di consegna. Con sede a Philadelphia, PA e con operazioni di dati a Phoenix, AZ, SportsDataIO attualmente serve più di 200 clienti che vanno dalle startup alle organizzazioni aziendali.

[About you SportsDataIO \(https://sportsdata.io/company\)](https://sportsdata.io/company)

Free Trial

SportsDataIO mette a disposizione un servizio di API gratuito cui è possibile accedere seguendo la documentazione



Developer Portal

Getting Started

Implementation Guide

Aggregated Odds Guide

Widgets Guide

+ Coverages

+ API Documentation

+ Data Dictionary

+ Integration Resources

+ Content

FAQ

Getting Started with Sports Data APIs

At SportsDataIO, we maintain the most robust [sports data feeds](#) in the industry. In order to start testing the integration, the first thing you'll need is an API subscription key. If you have not already received a key from our [sales team](#), you can get one by signing up for a free trial using the links below. If you're working with a team of developers, we'd recommend making a generic password that you can send to your developers, so they can also login to the same account, and have access to API subscription keys.

Choose the sport that you'd like a free trial for:

NFL	PGA
MLB	NASCAR
NBA	NCAA Football
NHL	NCAA Basketball
Soccer	League of Legends
	CS:GO

Per cominciare è necessario scegliere una lega: in questo tutorial verranno analizzati i dati relativi al campionato di basket americano NBA.

Cliccando su NBA verremo reindirizzati al seguente link <https://sportsdata.io/cart/free-trial/nba> (<https://sportsdata.io/cart/free-trial/nba>)

sportsdata.io SPORTS DATA SOLUTIONS DEVELOPERS CONTACT US 610.233.1333 MY ACCOUNT LOGOUT

NBA API Free Trial

NBA

Contact Information
We use your contact information to deliver your order.

First Name: Giovanni Last Name: Imbesi

Company: Individual Phone: Phone

Email: imbesigiovanni00@gmail.com

☐ Yes, keep me informed of future news and product updates

☐ Yes, I agree to the [Terms of Service](#) and [Privacy Policy](#)

SUBMIT

Da qui procederemo alla creazione di un account gratuito.

API KEY

Una volta creato, nella sezione Subscription, sarà possibile vedere l' API KEY associata al nostro account, tramite la quale ottenere i dati messi a disposizione dalla compagnia per la lega scelta.

sportsdata.io MY ACCOUNT DOWNLOADS **SUBSCRIPTIONS** CHANGE PASSWORD MANAGE CREDIT

My Subscriptions

Below are your current subscriptions.

FantasyData & BettingData **SportsDataIO**

NBA API Free Trial

● **Active**

Start Date: November 10, 2021

Exp Date: None

API Keys: 8adae53fad894151a48bb596dd3af164

LAUNCH DEVELOPER PORTAL

CANCEL SUBSCRIPTION

Cliccando su "Launch Developer Portal" si aprirà il portale di accesso con varie informazioni

relative l'NBA.

Developer Portal

- Getting Started
- Implementation Guide
- Aggregated Odds Guide
- Widgets Guide
- + Coverages
- API Documentation
 - + NFL
 - + MLB
 - NBA
 - + Sports Data
 - + News & Images
 - + Fantasy Data
 - + Betting Data
 - Miscellaneous
 - + NHL
 - + NCAA Football
 - + NCAA Basketball
 - + Golf
 - + NASCAR
 - + Soccer
 - + MMA
 - + WNBA
 - + NCAA Women's Basketball
 - + LoL
 - + CS:GO
- + Data Dictionary
- + Integration Resources
- + Content
- FAQ

API Documentation

This is the documentation for SportsDataIO's NBA API. All of our API endpoints can be accessed via an **HTTP GET** request using your API key. The API key can be passed either as a query parameter or using the following HTTP request header.

Ocp-Apim-Subscription-Key: {key}

Search endpoints...

Are Games In Progress

Tags: Schedules & Game Day Info

Returns **true** if there is at least one game being played at the time of the request or **false** if there are none.

Endpoint: <https://api.sportsdata.io/v3/nba/scores/json/AnyGamesInProgress>

Recommended Call Interval: 5 Seconds

API Key

8adae53fad894151a48bb596dd3af164

Request Parameters

Name	Value	Type	Required	Description
format	json	string	true	Desired response format. Valid entries are XML or JSON .

Games by Date

Tags: Game State, Schedules & Game Day Info, Scores

Endpoint: <https://api.sportsdata.io/v3/nba/scores/json/GamesByDate/{date}>

Recommended Call Interval: 5 Seconds

API Key

8adae53fad894151a48bb596dd3af164

Request Parameters

Name	Value	Type	Required	Description
format	json	string	true	Desired response format. Valid entries are XML or JSON .
date		string	true	The date of the game(s). Examples: 2015-JUL-31 , 2015-SEP-01 .

Non sarà possibile accedere ai dati in mancanza di una API KEY: una volta completata la registrazione ed effettuato il login con il proprio account, questa apparirà già inserita. Per ottenere i dati è sufficiente realizzare delle richieste via HTTP GET, usando la propria chiave, verso gli endpoint forniti dal portale.

Per la nostra analisi ci limiteremo ad utilizzare alcune librerie fondamentali per la gestione dei dati:

- la libreria requests ci permetterà di realizzare le richieste HTTP GET;
- pandas è una libreria essenziale per la creazione e gestione dei database;
- matplotlib è utile per la rappresentazione dei dati mediante dei grafici
- display da IPython, usata per poter rappresentare le immagini

```
In [1]: import requests
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure as fg
from IPython import display
```

Per prima cosa, è necessario ottenere i dati relativi ai giocatori e quindi ai loro salari. In particolare SportsDataIO mette a disposizione i suoi dati in formato json.

```
In [2]: urlPlayers= "https://api.sportsdata.io/v3/nba/scores/json/Players?key
rPlayers=requests.get(urlPlayers)
jsonPlayers= rPlayers.json()
jsonPlayers
```

Out[2]:

```
[{'PlayerID': 20000441,
  'SportsDataID': '',
  'Status': 'Active',
  'TeamID': 1,
  'Team': 'WAS',
  'Jersey': 3,
  'PositionCategory': 'G',
  'Position': 'SG',
  'FirstName': 'Bradley',
  'LastName': 'Beal',
  'Height': 75,
  'Weight': 207,
  'BirthDate': '1993-06-28T00:00:00',
  'BirthCity': 'St. Louis',
  'BirthState': 'MO',
  'BirthCountry': 'USA',
  'HighSchool': None,
  'College': 'Florida',
```

Inseriamo adesso i dati ottenuti in un dataframe.

```
In [3]: df=pd.DataFrame(jsonPlayers)
df.head()
```

Out[3]:

	PlayerID	SportsDataID	Status	TeamID	Team	Jersey	PositionCategory	Position	FirstN
0	20000441		Active	1	WAS	3.0	G	SG	Br
1	20000442		Active	22	HOU	1.0	G	PG	
2	20000443		Active	26	GS	32.0	F	SF	
3	20000452		Active	23	NO	41.0	G	SG	Ga
4	20000453		Active	5	ORL	31.0	G	SG	Terr

5 rows × 49 columns

Ottenuto il dataframe, possiamo procedere con le nostre analisi. Per cominciare, vogliamo trovare il giocatore più pagato della NBA e mostrarne una foto.

```
In [4]: urlPhoto=df.loc[df['Salary'].idxmax()]['PhotoUrl']
bestPlayerStats=(df.loc[df['Salary'].idxmax()])
print(bestPlayerStats[['FirstName','LastName', 'Position','Salary']])
from IPython import display
display.Image(urlPhoto, width=100, height=100)
```

FirstName Stephen
LastName Curry

Out[4]:



L'NBA è una lega le cui squadre sono suddivise in due divisioni: la Eastern e la Western conference. Determiniamo la conference più costosa, ovvero quella le cui squadre hanno, in totale, un monte ingaggi maggiore. Per raggiungere il nostro scopo non sono necessarie tutte le informazioni fornite dall'API, dunque, selezioniamo solo gli attributi fondamentali

```
In [5]: dbPlayers=df[['PlayerID', 'Team', 'Salary']].copy()  
dbPlayers.head()
```

Out[5]:

	PlayerID	Team	Salary
0	20000441	WAS	33724200.0
1	20000442	HOU	44310840.0
2	20000443	GS	1669178.0
3	20000452	NO	4910000.0
4	20000453	ORL	12500000.0

L'idea è quella di utilizzare una group by sulle squadre, al fine di ottenere per ognuna di esse, il totale degli stipendi dei rispettivi giocatori.

```
In [6]: dbSalaryTeams=pd.DataFrame(dbPlayers.groupby(by=["Team"], as_index=False)  
del dbSalaryTeams['PlayerID']  
dbSalaryTeams
```

Out[6]:

	Team	Salary
0	ATL	134732368.0
1	BKN	167465516.0
2	BOS	141588609.0
3	CHA	106223133.0
4	CHI	135195386.0
5	CLE	131976438.0
6	DAL	118605280.0
7	DEN	134808541.0
8	DET	79677933.0
9	GS	175192325.0
10	HOU	123147764.0

	Team	Salary
11	IND	132945873.0
12	LAC	165150600.0
13	LAL	150275652.0
14	MEM	97935225.0
15	MIA	132737483.0
16	MIL	146075260.0
17	MIN	135782463.0
18	NO	114929039.0
19	NY	141997352.0
20	OKC	51379042.0
21	ORL	115482329.0
22	PHI	141660440.0
23	PHO	131593332.0
24	POR	136767728.0
25	SA	99508279.0
26	SAC	130056334.0
27	TOR	135687580.0
28	UTA	153048231.0

Dobbiamo ora ottenere i dati relativi alle squadre, al fine di determinare la conference cui appartengono. Per farlo sfruttiamo un altro URL fornitoci da SportsDataIO, il quale comprende diversi dati sui Team NBA.

```
In [7]: url2="https://api.sportsdata.io/v3/nba/scores/json/teams?key=8adae531
rTeams=requests.get(url2)
jsonTeams=rTeams.json()
jsonTeams
```

Out[7]:

```
[{'TeamID': 1.
```

Estraiamo solo i dati relativi alla conference e la feature KEY del team. Per una più facile gestione dei due dataframe, andiamo a rinominare la colonna 'Key' come 'Team'.

```
In [8]: df2=pd.DataFrame(jsonTeams)
dbTeams=df2[['Key', 'Conference']].copy()
dbTeams=dbTeams.rename(columns={'Key': 'Team'})
dbTeams
```

Out[8]:

	Team	Conference
0	WAS	Eastern
1	CHA	Eastern
2	ATL	Eastern
3	MIA	Eastern
4	ORL	Eastern
5	NY	Eastern
6	PHI	Eastern
7	BKN	Eastern
8	BOS	Eastern
9	TOR	Eastern
10	CHI	Eastern
11	CLE	Eastern
12	IND	Eastern
13	DET	Eastern
14	MIL	Eastern
15	MIN	Western
16	UTA	Western
17	OKC	Western
18	POR	Western
19	DEN	Western
20	MEM	Western
21	HOU	Western
22	NO	Western
23	SA	Western
24	DAL	Western
25	GS	Western
26	LAL	Western
27	LAC	Western
28	PHO	Western
29	SAC	Western

Ottenuti e formattati i nostri dataframe, ci limitiamo ad effettuare una join sulla key 'Team', al fine di associare ad ogni squadra la corrispettiva Conference.

```
In [9]: dbMerged=pd.merge(dbTeams, dbSalaryTeams)  
dbMerged
```

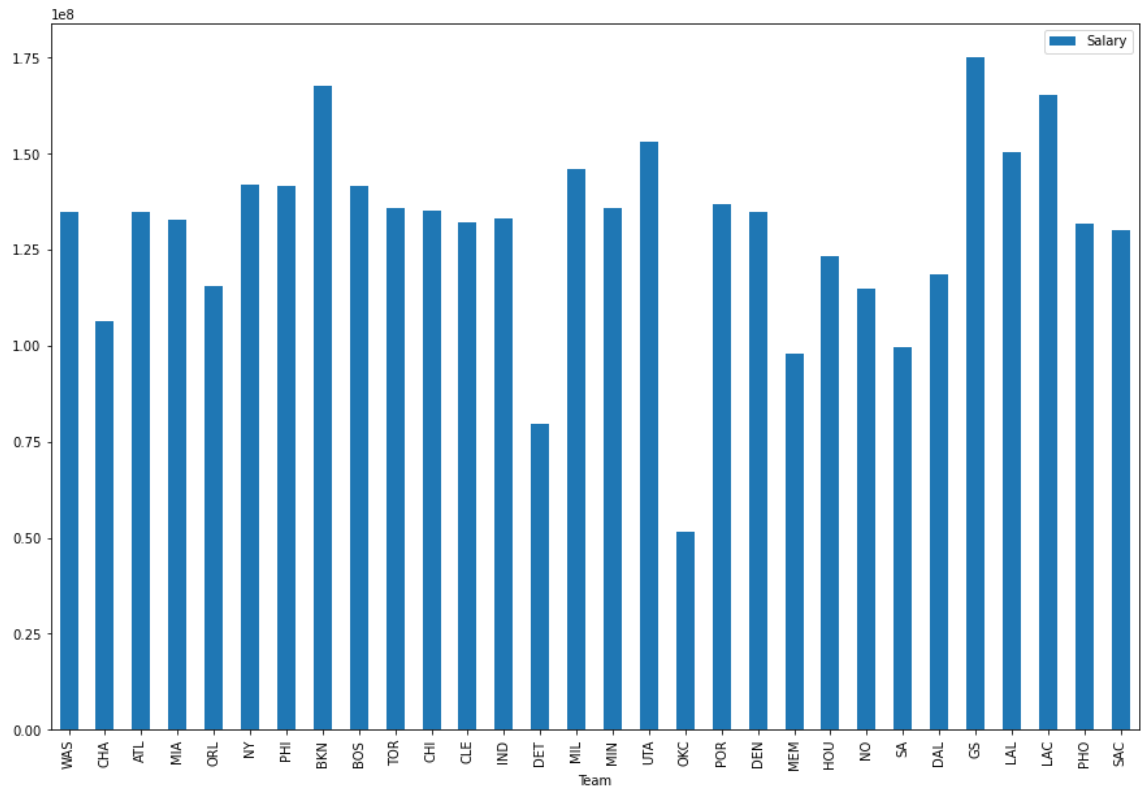
Out[9]:

	Team	Conference	Salary
0	WAS	Eastern	134736457.0
1	CHA	Eastern	106223133.0
2	ATL	Eastern	134732368.0
3	MIA	Eastern	132737483.0
4	ORL	Eastern	115482329.0
5	NY	Eastern	141997352.0
6	PHI	Eastern	141660440.0
7	BKN	Eastern	167465516.0
8	BOS	Eastern	141588609.0
9	TOR	Eastern	135687580.0
10	CHI	Eastern	135195386.0
11	CLE	Eastern	131976438.0
12	IND	Eastern	132945873.0
13	DET	Eastern	79677933.0
14	MIL	Eastern	146075260.0
15	MIN	Western	135782463.0
16	UTA	Western	153048231.0
17	OKC	Western	51379042.0
18	POR	Western	136767728.0
19	DEN	Western	134808541.0
20	MEM	Western	97935225.0
21	HOU	Western	123147764.0
22	NO	Western	114929039.0
23	SA	Western	99508279.0
24	DAL	Western	118605280.0
25	GS	Western	175192325.0
26	LAL	Western	150275652.0
27	LAC	Western	165150600.0
28	PHO	Western	131593332.0
29	SAC	Western	130056334.0

Potrebbe essere interessante capire graficamente qual è la squadra col monte ingaggi più

alto. Per farlo possiamo sfruttare il metodo plot.

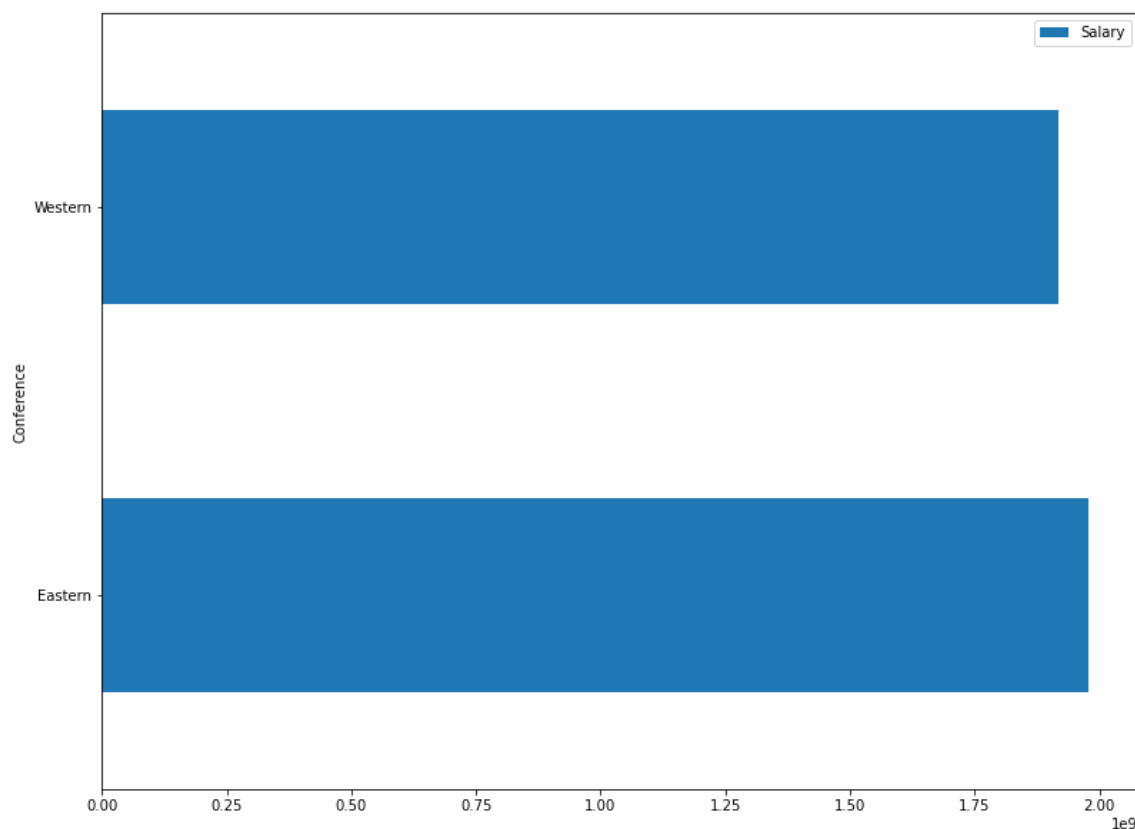
```
In [10]: plt.rcParams['figure.figsize'] = [15, 10]
dbMerged.plot(x='Team', y = 'Salary', kind='bar')
plt.show()
```



Infine, effettuando una group by sull'attributo Conference ed una sum, otteniamo il monte ingaggi totale per entrambe le Conference.

```
In [11]: print(dbMerged.groupby(by='Conference', as_index=False).sum())
dbConference=dbMerged.groupby(by='Conference', as_index=False).sum()
plt.rcParams['figure.figsize'] = [13, 10]
dbConference.plot(x='Conference', y = 'Salary', kind='barh')
plt.show()
```

	Conference	Salary
0	Eastern	1.978182e+09
1	Western	1.918180e+09



Considerazioni

Dal grafico risulta evidente che la Eastern Conference è quella più costosa, nonostante la differenza non sia particolarmente evidente, merito delle politiche economiche della NBA che cerca di garantire dunque un certo equilibrio.

Conclusione

Le API messe a disposizione da SportsDataIO sono di facile utilizzo e comprensione, ma soprattutto permettono di accedere ad una buona quantità di dati relativi non solo l'NBA ma anche altri sport. I database vengono costantemente aggiornati, ed il formato in cui vengono condivise le informazioni consente di analizzarle fin da subito senza particolari problemi di conversione o formattazione.