

```
import pandas as pd
import requests

import requests

url = "https://mashape-community-urban-dictionary.p.rapidapi.com/define"

querystring = {"term":"wat"}

headers = {
  "X-RapidAPI-Key": "06e1cc75edmshe6288875c7206afp171bb3jsn527ad23fad3e",
  "X-RapidAPI-Host": "mashape-community-urban-dictionary.p.rapidapi.com"
}

response = requests.get(url, headers=headers, params=querystring)

print(response.json())

{
  "list": [
    {
      "definition": "The only [proper] [response] to something that makes absolutely [no sense].",
      "permalink": "http://w...
```

data = pd.DataFrame(response.json()['list'])[['definition','permalink','thumbs_up','defid','word','author','written_on','example']]

data

	definition	permalink	thumbs_up	defid	word	author	written_on	example
0	The only [proper] [response] to something that...	http://wat.urbanup.com/3322419	3985	3322419	wat	watwat	2008-09-04T09:15:08.000Z	1: If all the animals on the [equator] were ca...
1	Slang. Indicates [confusion] or [a need] for ...	http://wat.urbanup.com/1308686	1218	1308686	wat	stewpidwhytemayun	2005-06-08T23:20:56.000Z	1: "[lolz] Yea sex is great and everything but...
2	[Wat] is the internet troll equivalent to wha...	http://wat.urbanup.com/5112312	214	5112312	wat	Razgriz117	2010-07-22T15:55:51.000Z	Guy 1: Hey, [Jimmy]. Can I ask you a question?...
3	A word, that when perfectly placed, is a	http://wat.urbanup.com/3351192	360	3351192	wat	Nevandal	2008-09-20T09:20:12.000Z	[Reno]: this chick looks kinda like

```
import requests

url = "https://matchilling-chuck-norris-jokes-v1.p.rapidapi.com/jokes/random"

headers = {
  "accept": "application/json",
  "X-RapidAPI-Key": "06e1cc75edmshe6288875c7206afp171bb3jsn527ad23fad3e",
  "X-RapidAPI-Host": "matchilling-chuck-norris-jokes-v1.p.rapidapi.com"
}

response = requests.get(url, headers=headers)

print(response.json())

{
  "categories": [],
  "created_at": "2020-01-05 13:42:19.576875",
  "icon_url": "https://assets.chucknorris.host/img/avatar/chuck-n...
```

```
import requests

url = "https://free-nba.p.rapidapi.com/players"

querystring = {"page":"0","per_page":"25"}

headers = {
  "X-RapidAPI-Key": "06e1cc75edmshe6288875c7206afp171bb3jsn527ad23fad3e",
  "X-RapidAPI-Host": "free-nba.p.rapidapi.com"
}



response = requests.get(url, headers=headers, params=querystring)

print(response.json())

{
  "data": [
    {
      "id": 46391370,
      "first_name": "Jake",
      "height_feet": None,
      "height_inches": None,
      "last_name": "Carter",
      "position":
```

```
data1 = pd.DataFrame(response.json()['data'])[['id', 'first_name', 'height_feet', 'height_inches', 'last_name', 'position']]
```

```
data1.head(10)
```

	id	first_name	height_feet	height_inches	last_name	position	
0	46391370	Jake	NaN	NaN	Carter		
1	2144	Rakeem	NaN	NaN	Christmas		
2	46391383	Bill	NaN	NaN	Closs		
3	122	Angel	6.0	10.0	Delgado	C	
4	123	Matthew	6.0	4.0	Dellavedova	G	
5	2819	Fennis	NaN	NaN	Dembo		
6	149	Keenan	NaN	NaN	Evans	G	
7	2299	Larry	NaN	NaN	Demic		
8	124	Luol	6.0	9.0	Deng	F	
9	2608	Kenny	NaN	NaN	Dennard		