

Find the 'id' of the movie "Andhadhun" using TMDb API.

```
In [1]: import requests
import json

In [34]: data = {'api_key': '4150bd5ca9e960f527d69c5701045bc0', 'query': 'Andhadhun'}
response = requests.get('https://api.themoviedb.org/3/search/movie', params = data)
data = response.json()
l = data['results']
for i in l:
    print(i['id'])

534780
```

Fetch the company id company 'Marvel Studios' using TMDb. Print the id.

```
In [40]: data = {'api_key': '4150bd5ca9e960f527d69c5701045bc0', 'query': 'Marvel Studios'}
response = requests.get('https://api.themoviedb.org/3/search/company', params = data)
data = response.json()
l = data['results']
for i in l:
    if i['name']== 'Marvel Studios':
        print(i['id'])

420
```

Find the vote count and vote average of the movie "3 Idiots" using the TMDb API

```
In [44]: data = {'api_key': '4150bd5ca9e960f527d69c5701045bc0', 'query': '3 Idiots'}
response = requests.get('https://api.themoviedb.org/3/search/movie', params = data)
data = response.json()
l = data['results']
for i in l:
    print(i['vote_count'], i['vote_average'])

2117 7.993
```

Fetch the names of top 5 similar movies to 'Inception' from the TMDb API.

```
In [1]: import requests

data = {'api_key': '4150bd5ca9e960f527d69c5701045bc0', 'query': 'Inception'}
response = requests.get('https://api.themoviedb.org/3/search/movie', params=data)
data = response.json()
for i in data['results']:
    if i['title']=='Inception':
        idd = i['id']
r = requests.get('https://api.themoviedb.org/3/movie/27205/similar', params={'api_key': '4150bd5ca9e960f527d69c5701045bc0', 'query': 'Inception'})
d = r.json()
for i in d['results'][:5]:
    print(i['title'])

This Side of Heaven
U-571
Mahogany
Changeling
Ready to wear
```

Fetch the top rated english movies in the US region using the TMDb API. From the result, print the first 10 movies which have original language as english. Also print their genres.

```
In [94]: #https://api.themoviedb.org/3/movie/top_rated
data = {'api_key': '4150bd5ca9e960f527d69c5701045bc0', 'region': 'US'}
response = requests.get('https://api.themoviedb.org/3/genre/movie/list', params=data)
data = response.json()
genres={}
for i in data['genres']:
    genres[i['id']] = i['name']

r = requests.get('https://api.themoviedb.org/3/movie/top_rated', params={'api_key': '4150bd5ca9e960f527d69c5701045bc0', 'region': 'US'})
d = r.json()
count=0
for i in d['results']:
    if(i['original_language']=="en"):
        print(i['title'],end=" ")
        for j in i['genre_ids']:
            print(genres[j],end=" ")
        print()
        count+=1
        if(count==10):
            break

The GodFather - Drama, Crime,
The Shawshank Redemption - Drama, Crime,
The Godfather Part II - Drama, Crime,
Schindler's List - Drama, History, War,
12 Angry Men - Drama,
The Dark Knight - Drama, Action, Crime, Thriller,
The Green Mile - Fantasy, Drama, Crime,
Pulp Fiction - Thriller, Crime,
Spider-Man: Across the Spider-Verse - Action, Adventure, Animation, Science Fiction,
Forrest Gump - Comedy, Drama, Romance,
```

Find the name and birthplace of the present most popular person according to TMDb API.

```
In [ ]: # https://api.themoviedb.org/3/trending/person/{time_window}
data = {'api_key': '4150bd5ca9e960f527d69c5701045bc0'}
response = requests.get('https://api.themoviedb.org/3/person/popular', params=data)
data = response.json()
data = data['results'][0]
popularid = data['id']

r = requests.get('https://api.themoviedb.org/3/person/169337', params={'api_key': '4150bd5ca9e960f527d69c5701045bc0'})
d = r.json()
print(d['id'])
print(d['name'], "-", d['place_of_birth'])
```

Fetch the Instagram and Twitter handle of Indian Actress "Alia Bhatt" from the TMDb API.

```
In [9]: #https://api.themoviedb.org/3/search/person
data = {'api_key': '4150bd5ca9e960f527d69c5701045bc0', 'query': 'Alia Bhatt'}
response = requests.get('https://api.themoviedb.org/3/search/person', params=data)
data = response.json()
for i in data['results']:
    idd = i['id']

#https://api.themoviedb.org/3/person/{person_id}/external_ids
data = {'api_key': '4150bd5ca9e960f527d69c5701045bc0'}
response = requests.get('https://api.themoviedb.org/3/person/1108120/external_ids', params=data)
data = response.json()
print(data['instagram_id'], data['twitter_id'])

aliaabhatt Aliaa08
```

Fetch the names of the character played by Tom Cruise in the movies:

```
In [34]: data = {'api_key': '4150bd5ca9e960f527d69c5701045bc0', 'query': 'Tom Cruise'}
response = requests.get('https://api.themoviedb.org/3/search/person', params=data)
data = response.json()
print(data['results'][0]['id'])

#https://api.themoviedb.org/3/person/{person_id}/movie_credits
r = requests.get('https://api.themoviedb.org/3/person/500/movie_credits', params={'api_key': '4150bd5ca9e960f527d69c5701045bc0'})
d = r.json()
for i in d['cast']:
    if(i['title']=='Top Gun'):
        print(i['character'])
        break
for i in d['cast']:
    if(i['title']=='Mission: Impossible - Fallout'):
        print(i['character'])
        break
for i in d['cast']:
    if(i['title']=='Minority Report'):
        print(i['character'])
        break
for i in d['cast']:
    if(i['title']=='Edge of Tomorrow'):
        print(i['character'])
        break

500
Lt. Pete 'Maverick' Mitchell
Ethan Hunt
Chief John Anderton
Maj. William 'Bill' Cage
```

Did James McAvoy play a role in the movie Deadpool 2. Print Yes or No.

```
In [28]: data = {'api_key': '4150bd5ca9e960f527d69c5701045bc0', 'query': 'Deadpool 2'}
response = requests.get('https://api.themoviedb.org/3/search/movie', params=data)
data = response.json()
print(data['results'][0]['id'])

#https://api.themoviedb.org/3/person/{person_id}/movie_credits
r = requests.get('https://api.themoviedb.org/3/movie/983498/credits', params={'api_key': '4150bd5ca9e960f527d69c5701045bc0'})
d = r.json()
for i in d['cast']:
    if(i['name']=='James McAvoy'):
        print("Yes")

383498
Yes
```

Using the result obtained in previous question, find out if James McAvoy was credited for his role in movie Deadpool 2. Print Yes or No.

```
In [43]: data = {'api_key': '4150bd5ca9e960f527d69c5701045bc0', 'query': 'James McAvoy'}
response = requests.get('https://api.themoviedb.org/3/search/person', params=data)
data = response.json()
print(data['results'][0]['id'])

r = requests.get('https://api.themoviedb.org/3/person/5530/movie_credits', params={'api_key': '4150bd5ca9e960f527d69c5701045bc0'})
d = r.json()
for i in d['cast']:
    if(i['original_title']=='Deadpool 2'):
        a = i['character']
if "uncredited" in a:
    print("No")
else:
    print("Yes")

5530
No
```

Fetch the overview of the TV Show "FRIENDS" using TMDb API.

```
In [95]: data = {'api_key': '4150bd5ca9e960f527d69c5701045bc0', 'query': 'FRIENDS'}
response = requests.get('https://api.themoviedb.org/3/search/tv', params=data)
data = response.json()
print(data['results'][0]['id'])

#https://api.themoviedb.org/3/movie/{movie_id}
d = {'api_key': '4150bd5ca9e960f527d69c5701045bc0', 'query': 'FRIENDS'}
r = requests.get('https://api.themoviedb.org/3/tv/1668', params={'api_key': '4150bd5ca9e960f527d69c5701045bc0'})
d = r.json()
for result in data.get('results'):
    if result.get('name') == 'Friends':
        print(result.get('overview'))

1668
Six young people from New York City, on their own and struggling to survive in the real world, find the companionship, comfort and support they get from each other to be the perfect antidote to the pressures of life.
Friends is a short-lived kids-oriented drama that aired in the spring of 1979. The series, which was produced by Aaron Spelling and aired on ABC, starred Charlie Aiken, Jill Whelan, and Jarrod Johnson as three Southern Californian 11-year-olds. Karen Morrow also appeared. Only five one-hour episodes were produced before the series was cancelled.
```

Fetch the name and air date of S06E05 of the TV Show 'The Big Bang Theory' from TMDb API.

```
In [65]: data = {'api_key': '4150bd5ca9e960f527d69c5701045bc0', 'query': 'Big Bang Theory'}
response = requests.get('https://api.themoviedb.org/3/search/tv', params=data)
data = response.json()
idd = data['results'][0]['id']

#https://api.themoviedb.org/3/tv/{series_id}/season/{season_number}
r = requests.get('https://api.themoviedb.org/3/tv/1418/season/6/episode/5', params={'api_key': '4150bd5ca9e960f527d69c5701045bc0'})
d = r.json()
print(d['name'], "-", d['air_date'])

The Holographic Excitation - 2012-10-25
```

Fetch the trending TV Shows for the week from the TMDb API and print the taglines of the top 5 shows. If there is no tagline, print 'Empty' instead

```
In [70]: #https://api.themoviedb.org/3/trending/tv/{time_window}
data = {'api_key': '4150bd5ca9e960f527d69c5701045bc0'}
response = requests.get('https://api.themoviedb.org/3/trending/tv/week', params=data)
data = response.json()
idd = []
for i in data['results'][:5]:
    idd.append(i['id'])
for i in idd:
    r = requests.get(f'https://api.themoviedb.org/3/tv/{i}', params={'api_key': '4150bd5ca9e960f527d69c5701045bc0'})
    d = r.json()
    if(d['tagline']==""):
        print("Empty")
    else:
        print(d['tagline'])

Who do you trust?
Empty
We're back again! Again!
A boy fights... for "the right death."
All empires fall.
```

Print the names of all the TV shows to be aired today whose original language is english.

```
In [96]: import requests

api_key = "4f67d20b7a0873cfeb5447295260470b"
api_link=https://api.themoviedb.org/3
params={'language': 'en', 'api_key': api_key}
header={'accept': 'application/json'}
response=requests.get(api_link+'Cv/airing_today/', headers=header, params=params)
data=response.json()
#data.get('total_pages')
for i in range(1, n+1):
    params={'language': 'en', 'api_key': api_key, 'page': i}
    r=requests.get(api_link+'Cv/airing_today/', headers=header, params=params)
    d=r.json()
    for i in d['results']:
        if i['original_language']=='en':
            print(i['name'])

Secret Invasion
Emmerdale
Hollyoaks
Love Island
PBS NewsHour
Riverdale
General Hospital
Days of Our Lives
House Hunters
Shortland Street
EastEnders
Naked News
Home and Away
Watch What Happens Live with Andy Cohen
The Patrick Star Show
Teen Titans Go!
Coronation Street
Nancy Drew
LIVE with Kelly and Mark
MasterChef
grown-ish
Guy's Grocery Games
All Elite Wrestling: Dynamite
The ReidOut
Tyler Perry's Sistas
American Pickers
The View
The Real Housewives of Orange County
WWE's The Bump
Alex Wagner Tonight
The Wonder Years
Work It Out Wombats!
Shark Week
Court Cam
Dr. Pimple Popper
The Hardy Boys
Hey Yahoo!
Temptation Island
Utopia
The Great British Sewing Bee
Joanna Lumley's Spice Trail Adventure
To Catch a Smuggler
Teen Mom: The Next Chapter
Holmes Family Rescue
F2: Chasing the Dream
Hunted Australia
LA Fire & Rescue
The Bold and the Beautiful
Rob & Roshes Vs Gold Diggers
Magical Japanese
Human Footprint
The Big D
My Strange Arrest
Big Freedia Means Business
Midst
Animals Aboard with Dr. Harry
Royal Blood
Split Second
Magandang Dilag
Gruen
The Ring of the Rowel: San Antonio Spurs Docuseries
A Body in the Basement
DEAN FUJIOKA 'Stars of the Lid' SPECIAL
YHC: 2nd Generation
```

Count the number of males and females in the cast of "Money Heist" using the TMDb API.

```
In [84]: data = {'api_key': '4150bd5ca9e960f527d69c5701045bc0', 'query': 'Money Heist'}
response = requests.get('https://api.themoviedb.org/3/search/tv', params=data)
data = response.json()
idd = data['results'][0]['id']

#https://api.themoviedb.org/3/tv/{series_id}/credits 71446
r = requests.get('https://api.themoviedb.org/3/tv/71446/credits', params={'api_key': '4150bd5ca9e960f527d69c5701045bc0'})
d = r.json()
m=0
f=0
for i in d['cast']:
    if(i['gender']==2):
        m+=1
    else:
        f+=1
print(m,f)

10 5
```