

```
In [5]: #Importing libraries
```

```
import pandas as pd
import json
import os
```

```
In [6]: # Loading the json data
```

```
data_batch_dir='/Users/HVMS/Desktop/GitHub/CineAI/data'
```

```
In [10]: for sub_dir in sorted(os.listdir(data_batch_dir)):
          print(sub_dir) # Print each subdirectory in order
```

```
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
```

```
In [9]: for sub_dir in sorted(os.listdir(data_batch_dir+ '/2000/')):
          print(sub_dir)
```

```
batch_001.json
batch_002.json
batch_003.json
batch_004.json
batch_005.json
batch_006.json
batch_007.json
batch_008.json
```

```
In [15]: test_batch_1_2000 = data_batch_dir+ '/2000/batch_001.json'
          with open(test_batch_1_2000, 'r') as file_1:
              data_1 = json.load(file_1)
```

```
In [16]: len(data_1)
```

Out[16]: 1000

```
In [17]: #To get movie ID  
for movie_id in data_1:  
    for keys, values in movie_id.items():  
        print(keys)
```

98
8871
7443
9532
77
8358
22705
10867
4247
11688
10567
2123
2024
1359
1443
4234
3134
8843
5833
10501
1443
8358
10637
1900
10577
1636
9741
16366
107
2085
5994
5833
9600
2069
9679
11978
843
3536
20677
9383
9679
11978
9383
8584
1907
146
10559
10393
12107
3176
2085
1636
1493
10647
2655
59709

11518
10035
15999
19576
20677
10577
3536
16234
243
10481
10876
19457
71
10991
2155
9600
10783
36968
5551
2440
15653
49948
8452
641
2069
2067
5491
16
11075
9456
1597
134
889
4244
10571
55892
20770
8859
10461
12609
2332
10687
11973
21614
10571
5551
889
479
7290
20455
18079
8653
18041
14626
8452
49948

19419
1588
10641
105904
10471
24102
10873
193457
1588
30628
4958
101514
2787
31347
41245
447399
62182
392
10569
11366
12211
9871
105904
711
327
18011
10473
10862
8653
193457
16340
16187
117087
10385
8870
12480
10560
10597
18041
10429
11983
14805
10472
48246
13785
37588
60670
55888
10873
24102
1163
14181
919
10642
44321
44853

28031
11599
10562
21355
24100
41275
12650
24664
24476
10180
10480
30970
18079
19348
4226
6521
318
40466
27092
14736
10763
1837
9674
5001
2621
39468
20455
37302
2360
49979
29458
16225
31352
17908
18074
11119
112
258919
10688
30944
21531
16888
159666
11477
10383
11478
29572
1619
27181
29512
29015
12597
7290
19419
12619
27027

15489
40096
10416
9011
1242945
15319
14295
16990
742
37691
1698
336970
36301
17711
9791
4967
32834
1831
22597
16110
38047
75571
13930
10394
44203
92208
20993
13966
3178
89473
200995
19150
87867
9519
35650
10697
10394
258919
11119
12480
15489
52684
10391
30940
1609
18168
39468
8870
190794
18074
51333
21991
34729
65592
37722
2805

18168
51333
18292
196737
20637
22617
193932
244839
55101
131940
5257
34729
20697
98299
52654
197725
12509
26245
32250
36234
37722
39242
4973
50008
27711
27099
204839
44412
49291
48944
52684
50531
3716
7093
190794
44388
29270
14195
44379
23990
2805
32536
27320
14629
42332
60703
28346
43586
576359
87107
43555
10391
20067
64269
53105
415488

46462
19214
33347
25128
71887
36380
49135
27745
34899
27444
10305
12806
60587
21056
299024
65592
248058
92620
25014
23618
44490
71235
26450
45438
25014
23618
89473
248058
45438
26450
44490
348864
51531
61064
87867
60703
71235
207693
41656
31774
5247
44933
288116
2707
51107
32309
89297
207937
10205
19349
72753
41282
12724
21370
30940
106230

45418
44488
62127
350893
48328
41165
10499
157800
77519
45488
34027
261111
72356
31875
16275
45602
42877
9375
45511
85411
47595
166813
685908
46958
44756
21174
36047
444008
22426
35066
45861
51767
31053
33504
49514
47943
28745
25636
61064
291334
77921
45573
55296
221506
2089
110977
50380
15940
81277
125173
118727
29866
35064
77434
100607
6147

40249
26665
32567
81207
31443
267409
32067
44947
60074
23531
37998
322686
38132
3173
193526
39927
13898
43079
214680
47244
39056
174412
12255
18394
46406
10613
13899
9883
10558
60579
36273
10989
291312
39907
34856
196280
88494
120128
389437
49709
20152
19505
36246
195791
45043
17208
39880
50477
230562
27859
30440
38274
31610
348862
24470
187107

111244
25212
25520
62004
14273
34037
31776
55551
47191
9301
45389
51179
106813
218188
16404
72392
610133
45463
188032
407312
37047
45124
56666
48217
36753
15869
604999
54380
29937
75531
264117
9565
11925
54690
218183
52409
34704
19600
186315
57450
20883
33994
114162
64526
10469
17806
291726
203071
43592
36923
60119
25293
45548
77787
49453
195655

329269
27734
127849
1051942
75049
344225
61578
58195
21117
45490
38294
85214
108811
107277
52622
61849
28943
55921
428006
21842
291721
26436
153913
76177
51422
34299
75697
77576
167880
20438
106199
95136
276846
52790
90881
431229
213914
38135
166231
72261
51531
348864
36817
64266
61641
109051
52051
22470
30148
78452
52852
20276
45952
47724
77332
154299

496537
36044
103152
47302
207693
27540
72998
71115
15090
18778
89345
44387
302466
38503
211277
44047
25651
200419
433111
887789
40562
29136
69426
27742
58976
45905
5643
9892
271917
3218
43674
68572
28349
354674
122163
51861
68820
60452
171120
64357
874667
446774
62333
81760
89068
54986
124134
326924
273106
9995
49035
46119
37716
45055
41571
30809

74578
28988
637105
117262
44131
23751
273510
48123
29946
45123
186047
206163
5245
39498
206053
86008
10415
214246
72270
612521
201447
41425
71618
434965
76265
106234
75217
31920
66796
49785
31001
116790
416715
443716
205608
86781
217428
256104
115717
78962
113213
3152
140511
330698
126733
209084
100801
56419
203831
52833
66464
24104
24955
55026
49511
41644

357682
58062
289673
80959
281014
85265
265760
351196
450187
452440
15993
287807
175240
27736
395337
20445
41381
88660
99754
28127
62729
447511
60065
76819
188154
248904
44574
200919
256050
70773
30947
385502
50477
38274
51824
164123
191378
267569
281308
2394
115469
41738
62004
31610
477164
33901
49453
106813
25212
37047
352815
45389
348762
254981
447719
171106

72087
16874
276456
27336
64114
107975
99011
155666
110999
181056
128862
11303
178582
21620
598278
384093
31006
277073
34794
158431
55143
20585
71884
167087
71550
116011
343321
30780
408129
414353
21488
290537
32249
118170
423976
880144
254981
201447
11925
56419
302466
330443
302508
390948
259730
16673
218183
67252
160832
41644
264806
356083
252615
245934
48868
73455

67252
376730
922327
165573
37730
364475
235423
88007
15496
23655
73642
29941
96716
33901
453230
31321
321023
1021910
351208
156682
107312
55302
63607
107942
286922
693072
79704
45203
49947
81219
69659
123187
82348
40930
1024825
213867
74824
271667
23430
44845
267569
103688
261459
340017
284183
844452
641425
45302
72244
10492
166283
75953
60585
62124
155588
75864

109926
75778
279700
364264
191378
41738
1185915
160832
88910
221501
1184877
57704
51864
169801
82103
102242
42195
24908
390959
252356
701212
38390
38987
18153
477164
318602
360411
31778
203820
45474
194529
77620
83581
434134
345864
48868
366978
66159
214014
261561
64310
34056
315978
116688
201724
1107965
166899
112977
77908
73838
448324
39793
199747
83870
40010
1009586

104613
217175
182511
221841
229428
88634
265307
106490
77332
47724
25651
66796
45952
45302
78620
132393
208910
10561
31920
89345
152163
267512
44387
15090
426506
40390
2090
471732
352815
79080
196139
30717
114620
273410
427108
239547
209133
231193
89823
339308
103635
1067248
436039
49773
23537
27775
31140
161219

```
In [24]: # To get movie individual keys columns such as(endpoints) – expecting 15 endpoints
first_movie = data_1[0]
movie_id = list(first_movie.keys())[0]
endpoints = first_movie[movie_id].keys()

print(f"Endpoints for movie ID {movie_id}:")
```

```
for endpoint in endpoints:
    print(f"-- {endpoint}")
```

Endpoints for movie ID 98:

- details
- credits
- recommendations
- watch_providers

```
In [25]: # Getting keys from 'details' endpoint for first movie
first_movie = data_1[0] # Get first movie
movie_id = list(first_movie.keys())[0] # Get movie ID
details_keys = first_movie[movie_id]['details'].keys() # Get details keys

print(f"\nKeys in 'details' for movie {movie_id}:")
for key in sorted(details_keys):
    print(f"-- {key}")
```

Keys in 'details' for movie 98:

- genres
- id
- original_language
- overview
- popularity
- release_date
- runtime
- title

```
In [26]: # Getting the first movie and its ID
movie = data_1[0]
movie_id = list(movie.keys())[0]
movie_data = movie[movie_id]

# Print all endpoint keys
print(f"\nAll endpoint keys for movie ID {movie_id}:")
for endpoint in sorted(movie_data.keys()):
    print(f"\n{endpoint.upper()} keys:")
    try:
        keys = movie_data[endpoint].keys()
        for key in sorted(keys):
            print(f"-- {key}")
    except AttributeError:
        print("(No nested keys)")
```

All endpoint keys for movie ID 98:

CREDITS keys:

- cast
- crew

DETAILS keys:

- genres
- id
- original_language
- overview
- popularity
- release_date
- runtime
- title

RECOMMENDATIONS keys:

- results

WATCH_PROVIDERS keys:

- results

```
In [27]: movie = data_1[0]
movie_id = list(movie.keys())[0]
movie_data = movie[movie_id]

# Initializing counter
total_keys = 0

# To Count and print keys for each endpoint
print(f"\nKey counts for movie ID {movie_id}:")
for endpoint in sorted(movie_data.keys()):
    try:
        keys = movie_data[endpoint].keys()
        num_keys = len(keys)
        total_keys += num_keys
        print(f"{endpoint}: {num_keys} keys")
    except AttributeError:
        print(f"{endpoint}: No nested keys")

print(f"\nTotal number of keys across all endpoints: {total_keys}")
```

Key counts for movie ID 98:

credits: 2 keys
 details: 8 keys
 recommendations: 1 keys
 watch_providers: 1 keys

Total number of keys across all endpoints: 12

In []: