

```
In [1]: 1 import requests
        2 import json
        3 import pandas as pd
        4 from bs4 import BeautifulSoup
        5 import os
        6 import re
```

```
In [2]: 1 client_id = ████████████████████████████████████████████████████████████████████████████████
        2 client_secret = ██████████████████████████████████████████████████████████████████████████████████████
        3 token = ██████████████████████████████████████████████████████████████████████████████████████
        4 artist_name = 'taylor swift'
```

```

In [3]: 1 def get_artist_info(artist_name, page):
2         base_url = 'https://api.genius.com'
3         headers = {'Authorization': 'Bearer ' + token}
4         search_url = base_url + '/search?per_page=10&page=' + str(page)
5         data = {'q': artist_name}
6         response = requests.get(search_url, data=data, headers=headers)
7         return response
8
9         def get_song_url(artist_name, song_cap):
10            page = 1
11            songs = []
12
13            while True:
14                response = get_artist_info(artist_name, page)
15                json = response.json()
16
17                song_info = []
18                for hit in json['response']['hits']:
19                    if artist_name.lower() in hit['result']['primary_artist']['name'].lower():
20                        song_info.append(hit)
21
22                for song in song_info:
23                    if (len(songs) < song_cap):
24                        url = song['result']['url']
25                        songs.append(url)
26
27                if (len(songs) == song_cap):
28                    break
29                else:
30                    page += 1
31
32            print('Found {} songs by {}'.format(len(songs), artist_name))
33            return songs

```

Test the functions to get urls

```

In [ ]: 1 get_song_url(artist_name, 300)

```

Define function to get lyrics

```

In [4]: 1 def get_lyrics(url):
2         page = requests.get(url)
3         html = BeautifulSoup(page.text, 'html.parser')
4
5         title = html.find('title')
6         title_text = title.get_text()
7         title_text = title_text.encode('ascii', errors='ignore').decode()
8         song_title = title_text.replace("TaylorSwift ", "").replace(" Lyrics | Genius Lyrics", " ")
9
10        album = html.find('a', class_="PrimaryAlbum__Title-cuci8p-4 NcWGs")
11        try:
12            album_text = album.get_text()
13        except AttributeError:
14            album_title = 'unknown'
15        else:
16            album_title = album_text.encode('ascii', errors='ignore').decode()
17            album_title = re.sub('(20((0[1-9])|([1-9][0-9])))', '', album_title)
18            album_title = re.sub('[()]', '', album_title).strip()
19
20
21        lyrics = html.find_all('div', class_="Lyrics__Container-sc-lynbvzw-8 eOLwDW")
22        full_lyrics = ''
23        for section in lyrics:
24            current_section = section.get_text()
25            current_section = current_section.encode('ascii', errors='ignore').decode()
26            full_lyrics = full_lyrics + current_section
27        full_lyrics = re.sub(r'\\(\\[.]*?\\)\\]', '', full_lyrics)
28        full_lyrics = os.linesep.join([s for s in full_lyrics.splitlines() if s])
29
30        lyric_words = [word for word in full_lyrics.split(" ") if word != ""]
31        obj = {}
32        obj['song_title'] = song_title.lower()
33        obj['album'] = album_title.lower()
34        obj['lyrics'] = lyric_words
35        return obj

```

Test the function to get lyrics

```
In [5]: 1 lyrics = get_lyrics('https://genius.com/Taylor-swift-i-know-places-lyrics')
        2 print(lyrics)
```

...

Group song lyrics by album

```
In [6]: 1 albums = ["taylor swift", "fearless", "speak now", "red deluxe edition", "1989 deluxe", "reputation",
        2 album_song_dict = {}
        3 for album in albums:
        4     album_song_dict[album] = []
```

```
In [7]: 1 album_song_dict
```

```
Out[7]: {'taylor swift': [],
         'fearless': [],
         'speak now': [],
         'red deluxe edition': [],
         '1989 deluxe': [],
         'reputation': [],
         'lover': [],
         'folklore': [],
         'evermore': []}
```

Define and run function to get urls + lyrics, and write output to json files

```
In [8]: 1 def write_lyrics_to_file(artist_name, song_count):
2     all_songs = []
3     urls = get_song_url(artist_name, song_count)
4     for url in urls:
5         title_and_lyrics = get_lyrics(url)
6         all_songs.append(title_and_lyrics)
7
8         if title_and_lyrics['album'] in album_song_dict.keys():
9             album_song_dict[title_and_lyrics['album']].append(title_and_lyrics['lyrics'])
10
11     with open('lyrics_taylor_swift.json', 'w') as f:
12         json.dump(all_songs, f, indent=4)
13
14     with open("album_lyrics_1.json", 'w') as f:
15         json.dump(album_song_dict, f, indent=4)
16
17 write_lyrics_to_file(artist_name, 300)
```

...

Open json file with lyrics

```
In [9]: 1 with open('lyrics_taylor_swift.json', 'r') as f:
2     song_lyrics_obj = json.load(f)
3     song_lyrics_obj
```

...

Open json file with lyrics grouped by album

```
In [10]: 1 with open('album_lyrics_1.json', 'r') as f:
2     album_lyrics_obj = json.load(f)
3     album_lyrics_obj
```

...

Add correct albums for songs with mislabeled albums

```
In [11]: 1 for song in song_lyrics_obj:
2         if song['song_title'] == 'you need to calm down ':
3             album_lyrics_obj['lover'].append(song['lyrics'])
4         if song['song_title'] == 'lover ':
5             album_lyrics_obj['lover'].append(song['lyrics'])
6         if song['song_title'] == 'blank space ':
7             album_lyrics_obj['1989 deluxe'].append(song['lyrics'])
8         if song['song_title'] == '...ready for it? ':
9             album_lyrics_obj['reputation'].append(song['lyrics'])
10        if song['song_title'] == 'willow ':
11            album_lyrics_obj['evermore'].append(song['lyrics'])
12        if song['song_title'] == 'cruel summer ':
13            album_lyrics_obj['lover'].append(song['lyrics'])
14        if song['song_title'] == 'the archer ':
15            album_lyrics_obj['lover'].append(song['lyrics'])
16        if song['song_title'] == 'untouchable ':
17            album_lyrics_obj['fearless'].append(song['lyrics'])
18        if song['song_title'] == 'the man ':
19            album_lyrics_obj['lover'].append(song['lyrics'])
20        if song['song_title'] == 'bad blood ':
21            album_lyrics_obj['reputation'].append(song['lyrics'])
22        if song['song_title'] == 'miss americana & the heartbreak prince ':
23            album_lyrics_obj['lover'].append(song['lyrics'])
24        if song['song_title'] == 'london boy ':
25            album_lyrics_obj['lover'].append(song['lyrics'])
26        if song['song_title'] == 'cornelia street ':
27            album_lyrics_obj['lover'].append(song['lyrics'])
28        if song['song_title'] == 'wildest dreams ':
29            album_lyrics_obj['1989 deluxe'].append(song['lyrics'])
30        if song['song_title'] == 'daylight ':
31            album_lyrics_obj['lover'].append(song['lyrics'])
32        if song['song_title'] == 'death by a thousand cuts ':
33            album_lyrics_obj['lover'].append(song['lyrics'])
34        if song['song_title'] == 'i forgot that you existed ':
35            album_lyrics_obj['lover'].append(song['lyrics'])
36        if song['song_title'] == 'paper rings ':
37            album_lyrics_obj['lover'].append(song['lyrics'])
38        if song['song_title'] == 'false god ':
39            album_lyrics_obj['lover'].append(song['lyrics'])
40        if song['song_title'] == 'i think he knows ':
41            album_lyrics_obj['lover'].append(song['lyrics'])
42        if song['song_title'] == 'afterglow ':
```

```
43     album_lyrics_obj[ 'lover' ].append(song[ 'lyrics' ])
44     if song[ 'song_title' ] == "soon you'll get better ":
45         album_lyrics_obj[ 'lover' ].append(song[ 'lyrics' ])
46     if song[ 'song_title' ] == 'shake it off ':
47         album_lyrics_obj[ '1989 deluxe' ].append(song[ 'lyrics' ])
48     if song[ 'song_title' ] == 'its nice to have a friend ':
49         album_lyrics_obj[ 'lover' ].append(song[ 'lyrics' ])
50     if song[ 'song_title' ] == 'everything has changed ':
51         album_lyrics_obj[ 'red deluxe edition' ].append(song[ 'lyrics' ])
52     if song[ 'song_title' ] == 'mean ':
53         album_lyrics_obj[ 'speak now' ].append(song[ 'lyrics' ])
54     if song[ 'song_title' ] == 'fifteen ':
55         album_lyrics_obj[ 'fearless' ].append(song[ 'lyrics' ])
56     if song[ 'song_title' ] == 'sparks fly ':
57         album_lyrics_obj[ 'speak now' ].append(song[ 'lyrics' ])
58     if song[ 'song_title' ] == 'picture to burn ':
59         album_lyrics_obj[ 'taylor swift' ].append(song[ 'lyrics' ])
60     if song[ 'song_title' ] == 'the story of us ':
61         album_lyrics_obj[ 'speak now' ].append(song[ 'lyrics' ])
62     if song[ 'song_title' ] == 'ours ':
63         album_lyrics_obj[ 'speak now' ].append(song[ 'lyrics' ])
64     if song[ 'song_title' ] == 'tim mcgraw ':
65         album_lyrics_obj[ 'taylor swift' ].append(song[ 'lyrics' ])
66     if song[ 'song_title' ] == 'cold as you ':
67         album_lyrics_obj[ 'taylor swift' ].append(song[ 'lyrics' ])
68     if song[ 'song_title' ] == 'tied together with a smile ':
69         album_lyrics_obj[ 'taylor swift' ].append(song[ 'lyrics' ])
70     if song[ 'song_title' ] == 'the other side of the door ':
71         album_lyrics_obj[ 'fearless' ].append(song[ 'lyrics' ])
72     if song[ 'song_title' ] == 'jump then fall ':
73         album_lyrics_obj[ 'fearless' ].append(song[ 'lyrics' ])
74     if song[ 'song_title' ] == 'if this was a movie ':
75         album_lyrics_obj[ 'speak now' ].append(song[ 'lyrics' ])
76     if song[ 'song_title' ] == 'im only me when im with you ':
77         album_lyrics_obj[ 'taylor swift' ].append(song[ 'lyrics' ])
78     if song[ 'song_title' ] == 'superstar ':
79         album_lyrics_obj[ 'fearless' ].append(song[ 'lyrics' ])
80     if song[ 'song_title' ] == 'come in with the rain ':
81         album_lyrics_obj[ 'fearless' ].append(song[ 'lyrics' ])
82     if song[ 'song_title' ] == 'superman ':
83         album_lyrics_obj[ 'speak now' ].append(song[ 'lyrics' ])
84     if song[ 'song_title' ] == 'today was a fairytale ':
85         album_lyrics_obj[ 'fearless' ].append(song[ 'lyrics' ])
```

```
In [12]: 1 album_lyrics_obj
```

...

Resave json file with lyrics grouped by album

```
In [13]: 1 with open("album_lyrics_1.json", 'w') as f:
2         json.dump(album_lyrics_obj, f, indent=4)
```

Construct pandas dataframe

```
In [14]: 1 lyrics_df = pd.DataFrame(data=song_lyrics_obj)
2         lyrics_df
```

Out[14]:

	song_title	album	lyrics
0	cardigan	folklore	[Vintage, tee,, brand, new, phoneHigh, heels, ...
1	exile	folklore	[I, can, see, you, standing,, honeyWith, his, ...
2	lover	unknown	[We, could, leave, the, Christmas, lights, up,...
3	the 1	folklore	[I'm, doing, good,, I'm, on, some, new, shitBe...
4	look what you made me do	reputation	[I, don't, like, your, little, gamesDon't, lik...
...
295	check out this view	unreleased songs	[I, dont, need, a, lotBut, there's, one, thing...
296	my cure	unreleased songs	[I, know, we've, got, a, lot, to, sayBetween, ...
297	sunshine	unreleased songs	[I, walk, in, the, room, there's, a, beautiful...
298	i forgot that you existed (piano/vocal)	unknown	[So, the, clip, that, you're, about, to, hearl...
299	you're not sorry (taylor's version)	fearless taylors version	[All, this, time, I, was, wasting, hoping, you...

300 rows x 3 columns

Save dataframe to csv file for storage in PostgreSQL database.


```
In [15]: 1 lyrics_df.to_csv('lyrics_df.csv',index=False)
```