Base de données

Projet de Base de données – Spotify API
https://github.com/AmokraneMancer/project_bdd_spotify_api

Réalisé par :

MANCER Mohammed Amokrane

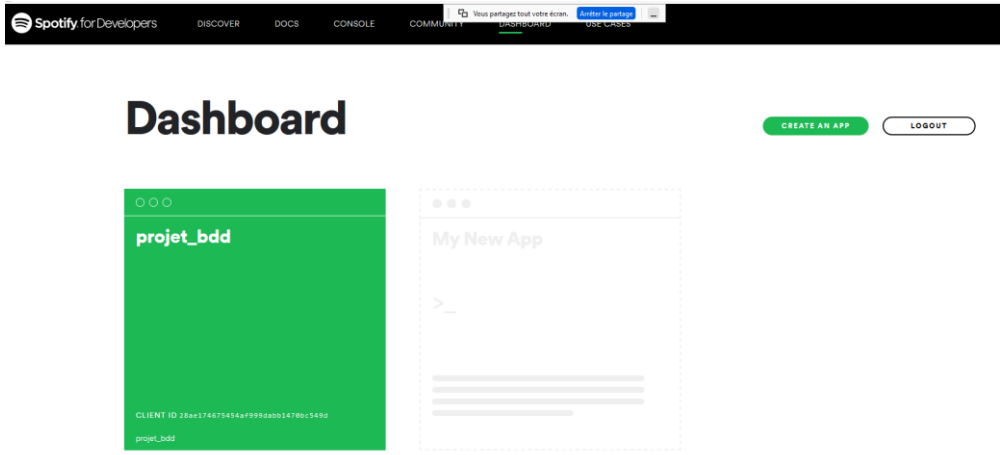
SADDEDINE Hassina

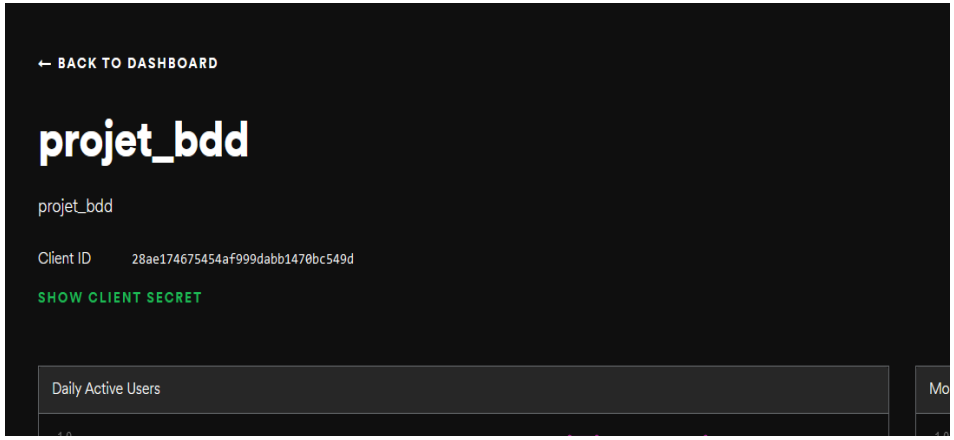
Présentation de projet

Le projet consiste à récupérer les données à partir d’une API publique et puis de les stocker dans une base de données relationnelle.

L’API de Spotify

Créer une application





Génération d'une clé (Token) pour se connecter à l'api

Entrée [1]:
```
import pyodbc
import requests
import pandas as pd
import json
```

Entrée [2]:
```
CLIENT_ID = '28ae174675454af999dabb1470bc549d'
CLIENT_SECRET = '7f05016f535e4260b7d3dcf771f14ecc'
AUTH_URL = 'https://accounts.spotify.com/api/token'

# POST
auth_response = requests.post(AUTH_URL, {
    'grant_type': 'client_credentials',
    'client_id': CLIENT_ID,
    'client_secret': CLIENT_SECRET,
})

# convert the response to JSON
auth_response_data = auth_response.json()

# save the access token
access_token = auth_response_data['access_token']
```

Entrée [3]:
```
access_token
```

Out[3]:
```
'BQDG3uPTyM9f0d8HGVGuZHVMvqnFNPdvZ-9v8Cq5i1qdyvsGhlov_clU8zdgNcTOGxnnc-mjVskjfCGphgY'
```

Entrée [4]:
```
headers = {
    'Authorization': 'Bearer {token}'.format(token=access_token)
}
```

Schéma entité-relation

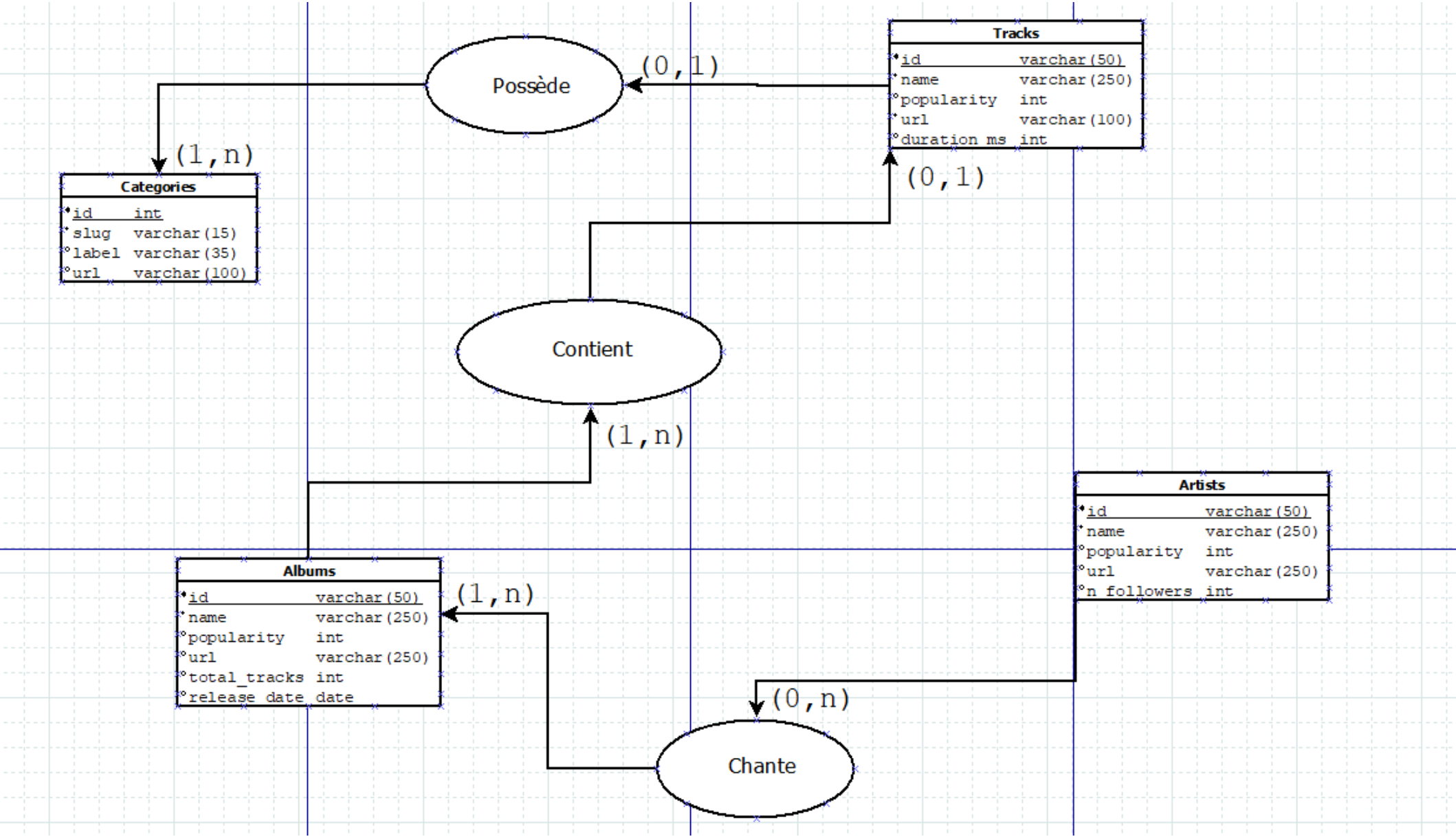


Schéma relationnel

Schéma initial non optimisé :

Tracks (id, name, popularity, url, duration_ms)

Artists (id, name, popularity, url, n_followers)

Albums (id, name, popularity, url, total_tracks, release_date)

Categories (id, slug, label, url)

Possède (#id_track, #id_cat)

Contient (#id_track, #id_album)

Chante (#id_artist, #id_album)

Schéma optimisé :

Tracks (id, name, popularity, url, duration_ms, #id_cat, #id_album)

Artists (id, name, popularity, url, n_followers)

Albums (id, name, popularity, url, total_tracks, release_date)

Categories (id, slug, label, url)

~~Possède (#id_track, #id_cat)~~

~~Contient (#id_track, #id_album)~~

Chante (#id_artist, #id_album)

1FN OK

2FN OK

3FN OK

Récupérer des catégories via ‘browse’

Get a List of Categories

Get a list of categories used to tag items in Spotify (on, for example, the Spotify player's "Browse" tab).

Endpoint

```
GET https://api.spotify.com/v1/browse/categories
```

Out[56]:

	href	icons	id	name
0	https://api.spotify.com/v1/browse/categories/t...	[{'height': 275, 'url': 'https://t.scdn.co/med...	toplists	Top Lists
1	https://api.spotify.com/v1/browse/categories/h...	[{'height': None, 'url': 'https://t.scdn.co/im...	holidays	Happy Holidays
2	https://api.spotify.com/v1/browse/categories/2020	[{'height': None, 'url': 'https://t.scdn.co/im...	2020	2020 Wrapped
3	https://api.spotify.com/v1/browse/categories/a...	[{'height': None, 'url': 'https://t.scdn.co/im...	at_home	At Home
4	https://api.spotify.com/v1/browse/categories/w...	[{'height': None, 'url': 'https://t.scdn.co/im...	wellness	Wellness
5	https://api.spotify.com/v1/browse/categories/r...	[{'height': None, 'url': 'https://t.scdn.co/im...	radar	RADAR
6	https://api.spotify.com/v1/browse/categories/mood	[{'height': 274, 'url': 'https://t.scdn.co/med...	mood	Mood
7	https://api.spotify.com/v1/browse/categories/pop	[{'height': 274, 'url': 'https://t.scdn.co/med...	pop	Pop
8	https://api.spotify.com/v1/browse/categories/h...	[{'height': 274, 'url': 'https://t.scdn.co/med...	hiphop	Hip Hop
9	https://api.spotify.com/v1/browse/categories/e...	[{'height': 274, 'url': 'https://t.scdn.co/med...	edm_dance	Dance/Electronic
10	https://api.spotify.com/v1/browse/categories/i...	[{'height': None, 'url': 'https://t.scdn.co/im...	indie_alt	Indie
11	https://api.spotify.com/v1/browse/categories/a...	[{'height': None, 'url': 'https://t.scdn.co/im...	alternative	Alternative

Catégories après nettoyage

t[78]:

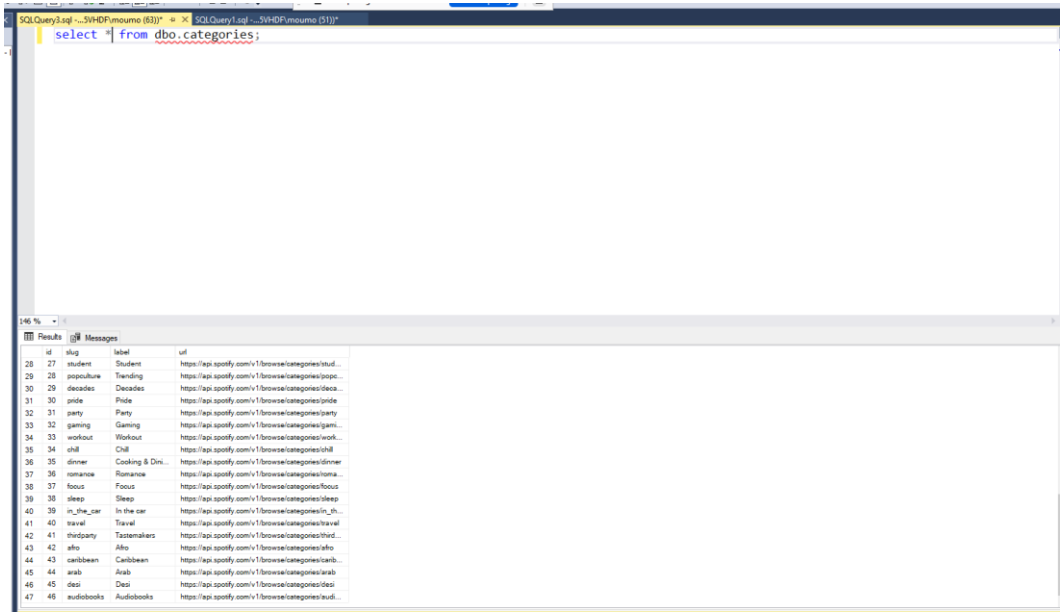
	url	slug	label
0	https://api.spotify.com/v1/browse/categories/t...	toplists	Top Lists
1	https://api.spotify.com/v1/browse/categories/h...	holidays	Happy Holidays
2	https://api.spotify.com/v1/browse/categories/2020	2020	2020 Wrapped
3	https://api.spotify.com/v1/browse/categories/a...	at_home	At Home
4	https://api.spotify.com/v1/browse/categories/w...	wellness	Wellness
5	https://api.spotify.com/v1/browse/categories/r...	radar	RADAR
6	https://api.spotify.com/v1/browse/categories/mood	mood	Mood
7	https://api.spotify.com/v1/browse/categories/pop	pop	Pop
8	https://api.spotify.com/v1/browse/categories/h...	hiphop	Hip Hop
9	https://api.spotify.com/v1/browse/categories/e...	edm_dance	Dance/Electronic
10	https://api.spotify.com/v1/browse/categories/i...	indie_alt	Indie

Insertion dans la base de donnés

```
.0]: con = pyodbc.connect('Driver={SQL Server};Server=DESKTOP-0G5VHDF\SQLEXPRESS;Database=Spotify;Trusted_Connection=yes;'
```

```
'9]: cursor = con.cursor()
# Insert Dataframe into SQL Server:
id = 0
for index, row in categories.iterrows():
    cursor.execute("INSERT INTO categories (id,slug,label,url) values(?,?,?,?)", id, row.slug, row.label, row.url)
    id = id + 1
con.commit()
cursor.close()
```

```
]: """
create table categories(
    id int not null,
    slug varchar(15),
    label varchar(35) not null,
    url varchar(100),
    PRIMARY KEY (id)
);
"""
```



	id	slug	label	url
28	27	student	Student	https://api.spotify.com/v1/browse/categories/stud...
29	28	popculture	Trending	https://api.spotify.com/v1/browse/categories/popc...
30	29	decades	Decades	https://api.spotify.com/v1/browse/categories/deca...
31	30	pride	Pride	https://api.spotify.com/v1/browse/categories/pride
32	31	party	Party	https://api.spotify.com/v1/browse/categories/party
33	32	gaming	Gaming	https://api.spotify.com/v1/browse/categories/gami...
34	33	workout	Workout	https://api.spotify.com/v1/browse/categories/work...
35	34	chill	Chill	https://api.spotify.com/v1/browse/categories/chill
36	35	dinner	Cooking & Dini...	https://api.spotify.com/v1/browse/categories/dinner
37	36	romance	Romance	https://api.spotify.com/v1/browse/categories/roma...
38	37	focus	Focus	https://api.spotify.com/v1/browse/categories/focus
39	38	sleep	Sleep	https://api.spotify.com/v1/browse/categories/sleep
40	39	in_the_car	In the car	https://api.spotify.com/v1/browse/categories/in_th...
41	40	travel	Travel	https://api.spotify.com/v1/browse/categories/travel
42	41	thirdparty	Tastemakers	https://api.spotify.com/v1/browse/categories/third...
43	42	afro	Afro	https://api.spotify.com/v1/browse/categories/afro
44	43	caribbean	Caribbean	https://api.spotify.com/v1/browse/categories/carib...
45	44	arab	Arab	https://api.spotify.com/v1/browse/categories/arab
46	45	desi	Desi	https://api.spotify.com/v1/browse/categories/desi
47	46	audiobooks	Audiobooks	https://api.spotify.com/v1/browse/categories/audi...

Récupérer les chansons et albums de chaque catégorie

`GET https://api.spotify.com/v1/search`

Request Parameters

Header Fields

HEADER FIELD	VALUE
Authorization	*Required.* A valid access token from the Spotify Accounts service: see the Web API Authorization Guide for details.

Query Parameters

QUERY PARAMETER	VALUE
q	*Required.* Search query keywords and optional field filters and operators. For example: `q=roadhouse%20blues`.
type	*Required.* A comma-separated list of item types to search across. Valid types are: `album`, `artist`, `playlist`, `track`, `show` and `episode`. Search results include hits from all the specified item types. For example: `q=name:abacab&type=album,track` returns both albums ***and*** tracks with "abacab" included in their name.

Get an Album

Get Spotify catalog information for a single album.

Endpoint

`GET https://api.spotify.com/v1/albums/{id}`

[6]:
```
# charger les catégories à partir de la base de données
con = pyodbc.connect('Driver={SQL Server};Server=DESKTOP-0G5VHDF\SQLEXPRESS;Database=Spotify;Trusted_Connection=yes;')
req = 'select slug from categories'
cursor = con.cursor()
cursor.execute(req)
categories_list = []
for row in cursor.fetchall():
    categories_list.append(row[0])
```

[8]:
```
response_list = {}
for i in range(len(categories_list)):
    category_name = categories_list[i]
    response_list[category_name] = []
    SEARCH_URL = 'https://api.spotify.com/v1/search?q={}&type=track&limit=50'.format(category_name)
    r = requests.get(SEARCH_URL, headers=headers)
    response_list[category_name].append(r.json())
```

duration_ms	explicit	external_ids	external_urls	href	id	is_local	name	popularity	preview_url	track_nu
200221	True	{'isrc': 'USSM12004501'}	{'spotify': 'https://open.spotify.com /track/6E...	https://api.spotify.com /v1/tracks /6EDO9iiTtwNv...	6EDO9iiTtwNv6waLwa1UUq	False	POPSTAR (feat. Drake)	90	https://p.scdn.co/mp3-preview /f06dde2517250354...	
166560	True	{'isrc': 'USQX91900309'}	{'spotify': 'https://open.spotify.com /track/6u...	https://api.spotify.com /v1/tracks /6uFn47ACjqYk...	6uFn47ACjqYkc0jADwEdj1	False	Pop Out (feat. Lil Tjay)	84	https://p.scdn.co/mp3-preview /505ae166c11669c6...	
227478	True	{'isrc': 'USAT22003620'}	{'spotify': 'https://open.spotify.com /track/2M...	https://api.spotify.com /v1/tracks /2MbdDtCv5LUV...	2MbdDtCv5LUVjYy9RuGTgC	False	WHATS POPPIN (feat. DaBaby, Tory Lanez & Lil W...	88	https://p.scdn.co/mp3-preview /3748f897bbea6068...	
220537	True	{'isrc': 'USEP41915005'}	{'spotify': 'https://open.spotify.com /track/4G	https://api.spotify.com /v1/tracks /4GssB27iJeqm	4GssB27iJeqmfGxS94Tfij	False	Popular Monster	79	https://p.scdn.co/mp3-preview /18707ef9fd54e5d0	

Insertion des chansons et albums

```
"""
create table tracks (
    id varchar(50),
    name varchar(250),
    popularity int,
    url varchar(250),
    id_album varchar(50),
    duration_ms int,
    cat_id int,
    FOREIGN KEY (cat_id) REFERENCES categories(id),
    CONSTRAINT PK_ID PRIMARY KEY NONCLUSTERED ([id], [cat_id], [name])
);
"""
```

```
"""
create table albums (
    id varchar(50) primary key,
    name varchar(250),
    popularity int,
    url varchar(250),
    total_tracks int,
    release_date date
);
```

```
for index, row in tracks.iterrows():
    try:
        album = row.album
        id_album = album['id']
        url = row.external_urls
        url = url['spotify']
        # l'album
        ALBUM_URL = 'https://api.spotify.com/v1/albums/{}'.format(id_album)
        r = requests.get(ALBUM_URL, headers=headers)
        json_albums = r.json()

        album_name = json_albums['name'] if 'name' in json_albums.keys() else 'null'
        album_popularity = json_albums['popularity'] if 'popularity' in json_albums.keys() else -1
        album_t_tracks = json_albums['total_tracks'] if 'total_tracks' in json_albums.keys() else -1
        album_date = json_albums['release_date'] if 'release_date' in json_albums.keys() else '2020-01-01'
        album_url = json_albums['external_urls']['spotify'] if 'external_urls' in json_albums.keys() else 'null'

        cursor.execute("INSERT INTO tracks (id , name, popularity, url, duration_ms, cat_id, album_id) values(?,?
                       row.id, row.label, row.popularity, url, row.duration_ms, cat_id, id_album)

        cursor.execute("INSERT INTO albums (id ,name,popularity,url, total_tracks, release_date) values(?,?,?,?,?
                       id_album, album_name, album_popularity, album_url, album_t_tracks, album_date )
```

SQLQuery2.sql -...5VHDF\moumo (72))* | tracks.sql - DESK...G5VHDF\moumo (67)) | SQLQuery1.sql -...5VHDF\moumo (66))*

```sql
select * from tracks
```

146 %

Results | Messages

	id	name	popularity	url	duration_ms	cat_id	album_id
1896	7wgQ9rl95fXtG7Lgts1sdP	End of All Hope - Remastered	29	https://open.spotify.com/track/7wgQ9rl95fXtG7Lgts1sdP	263397	29	5uEm5O3gMHJ1AZRQxSvDPV
1897	7wn8HSGhUiGREUBao34AJz	24 hours	27	https://open.spotify.com/track/7wn8HSGhUiGREUBao...	165459	22	0kZj3VuLwSfwFqUmL4coi3
1898	7wsOL6S0gabDU2OW95qISq	Chill AF	49	https://open.spotify.com/track/7wsOL6S0gabDU2OW9...	245989	34	2uLQpzNegtkOX2HnuQz7f
1899	7wsskVeImYaHhbcGEb7H8U	Romance	41	https://open.spotify.com/track/7wsskVeImYaHhbcGEb...	148009	36	4W9aVRGpS12uZahHzSZGRd
1900	7wz8LNfmGZ3C0PwKW9Ld...	Battery (Remastered)	57	https://open.spotify.com/track/7wz8LNfmGZ3C0PwKW...	312360	23	7CGhx63ODjdJqaBDVKc5j
1901	7wzLu2aumQh8yIdq5B55Fg	Dig Your Roots	50	https://open.spotify.com/track/7wzLu2aumQh8yIdq5B5...	214040	18	09KOjafTBRE28GVvXqkYC
1902	7x7m52J6wLQh9MZqdQkzx4	Student Memory	22	https://open.spotify.com/track/7x7m52J6wLQh9MZqd...	150834	27	6z7Fgo33tOVGEevlMchSsC
1903	7xNEqlq0BsFK0udSxh4BCW	Theme (from "The Patriot")	33	https://open.spotify.com/track/7xNEqlq0BsFK0udSxh4...	493600	29	7EgRKm3uO0CbW4Tqll2Rzf
1904	7xoLBXeXSSRpOFqoKlhiPb	Latino	38	https://open.spotify.com/track/7xoLBXeXSSRpOFqoKl...	61020	21	21SXoD0NMmD0PR1NkvfOlr
1905	7yoWLEP1GsNjVvcjawXz3z	Praise The Lord (Da Shine) (feat. Skepta)	80	https://open.spotify.com/track/7yoWLEP1GsNjVvcjawX...	205040	12	3MATDdrpHmQCmuOcozZjDa
1906	7yMLsyHDOlls3LjvEyolv4	Mind Warrior (Beta 13 Hz)	55	https://open.spotify.com/track/7yMLsyHDOlls3LjvEyolv4	179076	37	4EOJ0H7R7LEmsUzu72u9Bt
1907	7ynmAvlaIbYWpXtpX2PlpX	Dynamite - Instrumental	62	https://open.spotify.com/track/7ynmAvlaIbYWpXtpX2Pl...	198770	25	6K4chJALBBMYmXjwgvqahx
1908	7yQ1JstBnU0UNWA2jq9WPS	Herz und Mund und Tat und Leben, BWV 147: C...	17	https://open.spotify.com/track/7yQ1JstBnU0UNWA2jq9...	408683	45	2OEG3a5NScVMJhFjSnl4vL
1909	7yR5pFWmSjzHJIeQkgog4	ROCKSTAR (feat. Roddy Ricch)	93	https://open.spotify.com/track/7yR5pFWmSjzHJIeQkgo...	181733	12	623PL2MBq5
1910	7ywpz0H8zX3ASJfD49VnY	Workout, Pt. 1 (Through the Threshold)	25	https://open.spotify.com/track/7ywpz0H8zX3ASJfD49...	142308	33	58ju4h6P1ph
1911	7zKLsJNpMuVV870VX2rcKb	Herz und Mund und Tat und Leben, BWV 147: X:...	0	https://open.spotify.com/track/7zKLsJNpMuVV870VX2...	220373	13	09gLL9e64gy
1912	7zQRF2pG5pty9sck6L6hF4	Funky Little Beat	45	https://open.spotify.com/track/7zQRF2pG5pty9sck6L6...	313666	17	324QdGjl2W
1913	7zrhy9OhtcO7MdIvbbJiOa	December	72	https://open.spotify.com/track/7zrhy9OhtcO7MdIvbbJiOa	116266	34	6yi7FjzeCHG
1914	7zXKSgoiJfoLBsNmF2ls5V	Sigueme Y Te Sigo	0	https://open.spotify.com/track/7zXKSgoiJfoLBsNmF2ls5V	209786	21	0a1kxLoBkTe
1915	7zXteUOVPenxyFc4pTP6QB	Colorise	25	https://open.spotify.com/track/7zXteUOVPenxyFc4pTP...	125402	9	1AsAFGxSeZ

Query executed successfully.

SQLQuery2.sql -...5VHDF\moumo (72))* | tracks.sql - DESK...G5VHDF\moumo (67)) | SQLQuery1.sql -...5VHDF\moumo (66))*

```sql
select * from albums
```

146 %

Results | Messages

	id	name	popularity	url	total_tracks	release_date
1473	7vfuTRXIAYJz5Ue8SddnTr	Nightmare	73	https://open.spotify.com/album/7vfuTRXIAYJz5Ue8Sdd...	11	2010-07-23
1474	7vLxDionyOHSDmYS7Wpw...	null	-1	null	-1	2020-01-01
1475	7voOr4xZlOtUAz9vK1oaUM	Latin Rap Beat	30	https://open.spotify.com/album/7voOr4xZlOtUAz9vK1o...	1	2018-12-08
1476	7vyfRgqJvBcBezgEkAqKZQ	Data Punk	58	https://open.spotify.com/album/7vyfRgqJvBcBezgEkAq...	5	2020-03-07
1477	7wWQFNpOLyRK6y4Ch1Na...	null	-1	null	-1	2020-01-01
1478	7wzcIKvtXDWs1VA2BMHuaY	All We Got (feat. KIDDO)	77	https://open.spotify.com/album/7wzcIKvtXDWs1VA2BM...	1	2020-10-16
1479	7xI50xr9NDkd3i2kBbzsNZ	null	-1	null	-1	2020-01-01
1480	7xPV0IATNfqidql84mrG6u	Happy Endings	55	https://open.spotify.com/album/7xPV0IATNfqidql84mrG6u	6	2011-04-11
1481	7y4IXCVZyxKtzUjM8NNAEp	Nice Colors	62	https://open.spotify.com/album/7y4IXCVZyxKtzUjM8NN...	7	2018-05-17
1482	7y6INunWnPltlB58DAO6YB	Roots	29	https://open.spotify.com/album/7y6INunWnPltlB58DAO...	1	2020-09-30
1483	7yeVUGG3xvGjKwhM2FBlnZ	The Classic Years: 1956-1962	74	https://open.spotify.com/album/7yeVUGG3xvGjKwhM2...	62	1992-01-01
1484	7yKxxL9eQMJ4y3g5e8yU6C	Rockin' Around The Christmas Tree	57	https://open.spotify.com/album/7yKxxL9eQMJ4y3g5e8y...	1	2017-10-20
1485	7yILdI78Gbnb6OsMqM5g0J	null	-1	null	-1	2020-01-01
1486	7yTjzInNdNQJAIXAboi1nh	Making Movies	51	https://open.spotify.com/album/7yTjzInNdNQJAIXAboi1nh	7	1980-10-17
1487	7ywSMs3G2OWOGVKMqc7...	SALES LP	64	https://open.spotify.com/album/7ywSMs3G2OWOGVK...	15	2016-04-20
1488	7z4GhRfLqfSkqrj5F3Yt2B	Heaven Or Hell	85	https://open.spotify.com/album/7z4GhRfLqfSkqrj5F3Yt2B	12	2020-03-13
1489	7zBue2Vuzg4Z3ncRXaIkJg	Celebrity	56	https://open.spotify.com/album/7zBue2Vuzg4Z3ncRXaI...	13	2001-07-24
1490	7zf9kyxc3ZhJv79HrjTRir	Timmy Turner	63	https://open.spotify.com/album/7zf9kyxc3ZhJv79HrjTRir	1	2016-07-22
1491	7zjGWpumcMPpIlfjfi4qH3	null	-1	null	-1	2020-01-01
1492	7zqahK0WaZsYw7ghWVgp3n	Instrumental Coffee Jazz Moments	54	https://open.spotify.com/album/7zqahK0WaZsYw7ghW...	20	2019-03-25

Vous ... participants

pour revenir à l'appel vidéo lorsque ... arrêter de présenter

Récupérer et insérer les artistes

Get an Artist

Get Spotify catalog information for a single artist identified by their unique Spotify ID.

Endpoint

```
GET https://api.spotify.com/v1/artists/{id}
```

```
cursor = con.cursor()
album_by_artist = []
for album_id in albums_list:
    ALBUM_URL = 'https://api.spotify.com/v1/albums/{}'.format(album_id)
    r = requests.get(ALBUM_URL, headers=headers)
    if 'artists' in r.json().keys():
        artists = r.json()['artists']
    else:
        pass
    for ar in artists:
        try:
            artist_id = ar['id']
            ARTIST_URL = 'https://api.spotify.com/v1/artists/{}'.format(artist_id)
            artist_json = requests.get(ARTIST_URL, headers=headers).json()
            artist_name = artist_json['name']
            artist_popularity = artist_json['popularity']
            artist_url = artist_json['external_urls']['spotify']
            n_followers = artist_json['followers']['total']
            album_by_artist.append((album_id, artist_id))
            cursor.execute("INSERT INTO artists (id , name, popularity, url, n_followers) values(?,?,?,?,?)",
                                artist_id, artist_name, artist_popularity, artist_url, n_followers)
        except Exception as ex:
            sqlstate = ex.args[0]
            if sqlstate == '23000':
                pass

con.commit()
cursor.close()
```

SQLQuery2.sql -...5VHDF\moumo (72))* | tracks.sql - DESK...G5VHDF\moumo (67)) | SQLQuery1.sql -...5VHDF\moumo (66))*

```sql
select * from artists
```

146 %

Results | Messages

	id	name	popularity	url	n_followers
840	7nqOGRxlXj7N2JYbgNEjYH	SEVENTEEN	77	https://open.spotify.com/artist/7nqOGRxlXj7N2JYbgN...	4559017
841	7oMRcCu0OYSCtCyS3P37iC	Afro B	64	https://open.spotify.com/artist/7oMRcCu0OYSCtCyS3...	119190
842	7pcwhURfHXgfREAmsabGOh	Dinner MaMa	44	https://open.spotify.com/artist/7pcwhURfHXgfREAms...	587
843	7pFeBzX627ff0VnN6bxPR4	Designer	69	https://open.spotify.com/artist/7pFeBzX627ff0VnN6b...	3090204
844	7pTRluK9b4psmCPOf50uyc	Stream	28	https://open.spotify.com/artist/7pTRluK9b4psmCPOf5...	445
845	7pXu47GoqSYRajmBCjxdD6	Vulfpeck	64	https://open.spotify.com/artist/7pXu47GoqSYRajmBC...	592570
846	7qGx9nJ5gKY0tLgMRcNAqC	Rocky Brim	40	https://open.spotify.com/artist/7qGx9nJ5gKY0tLgMR...	17723
847	7qKxhFTGcJ7w7JEFGqqWwK	Piano Peace	64	https://open.spotify.com/artist/7qKxhFTGcJ7w7JEFG...	9032
848	7t0rwkOPGlDPEhaOcVtOt9	The Cranberries	75	https://open.spotify.com/artist/7t0rwkOPGlDPEhaOc...	2284073
849	7tYKF4w9nC0nq9CsPZTHyP	SZA	82	https://open.spotify.com/artist/7tYKF4w9nC0nq9CsP...	4055062
850	7tzJbwhbS6NHN9OvfMsn3o	Culture	28	https://open.spotify.com/artist/7tzJbwhbS6NHN9Ovf...	45
851	7udo52qeGxSsK69WMRmBvP	Jero Toga	46	https://open.spotify.com/artist/7udo52qeGxSsK69W...	564
852	7vfmvgP5tbJg8tMILY3OOq	Bossa Nova Jazz	52	https://open.spotify.com/artist/7vfmvgP5tbJg8tMILY3...	11601
853	7vgJnbva49FV9p5EkNlKq7	Baby Lullaby	58	https://open.spotify.com/artist/7vgJnbva49FV9p5EkN...	3375
854	7wyBxBXLVLaiFZjxHCpvTe	Koop	52	https://open.spotify.com/artist/7wyBxBXLVLaiFZjxHC...	125330
855	7xHEu7guf3Zt1a2pFqvOiM	Flatleyanita	45	https://open.spotify.com/artist/7xHEu7guf3Zt1a2pFqv...	726
856	7xUDsHk2yrRFkMGX7ZNnr4	Cliff Targum	15	https://open.spotify.com/artist/7xUDsHk2yrRFkMGX7...	31
857	7yCGrS6Xh3UngvY6Ad5sMJ	LEGADO 7	67	https://open.spotify.com/artist/7yCGrS6Xh3UngvY6A...	829536
858	7yPPzu5UdAK7yagQqjEZQm	Party Favor	62	https://open.spotify.com/artist/7yPPzu5UdAK7yagQqj...	146006
859	7zFBW2JxM4bgTTKxCRcS8Q	Hiphop Tamizha	64	https://open.spotify.com/artist/7zFBW2JxM4bgTTKxC...	1682588

Insérer dans la table chante (album_by_artist)

```
cursor = con.cursor()
for t in album_by_artist:
    try:
        cursor.execute("INSERT INTO album_by_artist (id_album , id_artist) values(?,?)",
                                    t[0], t[1])
    except Exception as ex:
            sqlstate = ex.args[0]
            if sqlstate == '23000':
                pass
con.commit()
cursor.close()
```

```
"""
create table track_in_album (
    id_track varchar(50),
    id_album varchar(50),
    FOREIGN KEY (id_track) REFERENCES tracks(id),
    FOREIGN KEY (id_album) REFERENCES albums(id),
    CONSTRAINT PK_idf PRIMARY KEY NONCLUSTERED
    ([id_track], [id_album])
);
"""
```