**Spotify API – Get Artist Genres, Select Pop Singers, Get Artist’s Albums**

* **目的**

透過Spotify API取得每一位歌手的類型，並篩選出流行音樂的歌手(去除背景音樂較複雜的歌手)，預先定義欲刪除的類型清單。

* **程式碼**

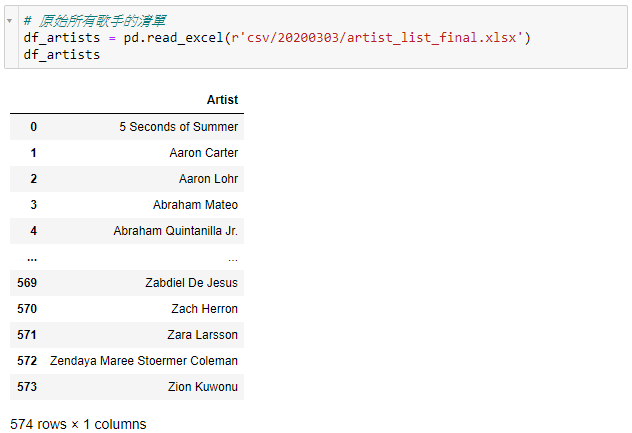
SpotifyAPI/20200303getArtistID+Genres.ipynb

SpotifyAPI/ 20200303getArtistAlbums.ipynb

* **步驟**

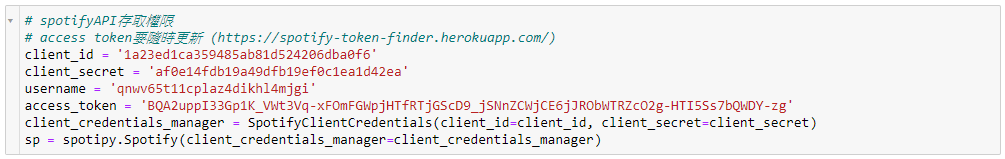
1. **匯入原始所有歌手的清單**

* 透過歌手網站(https://www.thefamouspeople.com/pop-singers.php)抓取流行音樂歌手，程式碼詳見SpotifyAPI/collectArtists.ipynb。



1. **spotifyAPI存取權限**

* 透過spotifyAPI開發者介面取得id和secret
* access token要隨時更新 (https://spotify-token-finder.herokuapp.com/)



1. **取得歌手資訊**

* 透過歌手名稱搜尋(輸入name)，取得歌手的名稱name、編號ID、連結url，三個資訊合併成artist\_info，若有歌手不存在則回傳check=1。



1. **取得歌手名稱和類型**

* 需使用到bearer\_token和headers，bearer\_token透過access\_token取得，headers透過定義'Accept'、'Content-Type'和'Authorization'。
* 透過歌手ID搜尋，回傳reponse為json格式。json格式中取得name(型態str)、id(型態str)和genres(型態list)。
* genres的list型態轉換：透過 ','.join() 手法，轉換成str型態。
* 參考：https://github.com/Leats/spotify-historymaker/blob/master/spotifyhistorymaker.py



1. **輸出歌手資訊的csv**

* 分別將每位歌手的id、name、genres分別存到3個list中，利用這3個list輸出csv。
* 為避免亂碼產生，設置encoding="utf\_8\_sig"。
* 第一次寫入使用"w" writer.writerow(['Artist','Id','Genres'])，第二次之後寫入使用"a"。
* 為避免一次資料量過大，分批執行寫入，設定變數base，一次處理100筆資料，base = 0 (df切割[base:base+100])，直到不滿100筆資料時，base = 500 (df切割[500:])。
* 透過步驟(3)回傳的check值判斷，是否存在此歌手，若不存在則不寫入csv。
* 輸出：csv/20200303/v2[版本資料夾]/artist\_list\_ID+Genres\_v2[版本].csv



1. **匯入要保留的歌手類型清單**

* 透過現有的網站資料爬蟲並整理篩選，詳見getArtistGenres.ipynb。
* Spotify歌手所有類型從<http://everynoise.com/everynoise1d.cgi?root=pop&scope=all>抓取，並把所有類型建立csv檔案(csv/20200303/artist\_genres.csv)，總共4009個類別。
* 匯入欲刪除類型的字串，須排除的類別檔案'csv/20200303/delete\_string.txt'，一行一行讀入，去除換行符號 ('\n') 並利用 '|'.join() 取得字串(string)。
* 篩選dataframe中含有特定字元的欄位，'|'進行多個條件篩選，因此才需使用'|'做join動作。

df2 = df.loc[~df ['欄位名稱'].str.contains(string欲刪除的字串)].reset\_index(drop=True)

* 輸出篩選後的檔案(csv/20200303/artist\_genres\_select.xlsx)，之後可直接匯入。



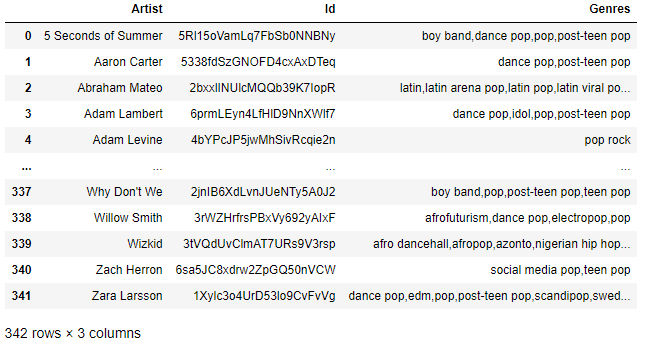
1. **讀取含有歌手name、id、genres的csv並進行前處理**

* 讀取步驟(5)產生的檔案「csv/20200303/v2[版本資料夾]/artist\_list\_ID+Genres\_v2[版本].csv」



* 刪除Genres為空的那一列(上圖中為NaN)，並輸出一份新的csv檔案，如下圖為篩選後的結果。

csv/20200303/v2[版本資料夾]/artist\_list\_ID+Genres\_drop\_v2[版本].csv



1. **逐行篩選歌手類別**

* 歌手的總類別清單：artist\_genres\_select.xlsx。(註：總類別為「不要」的歌手類型)
* 匯入步驟(7)的檔案csv/20200303/v2[版本資料夾]/artist\_list\_ID+Genres\_drop\_v2[版本].csv篩選。
* 把歌手類型清單的類型欄位轉換成list

教學：<https://www.interviewqs.com/ddi_code_snippets/get_list_pandas_datafra>

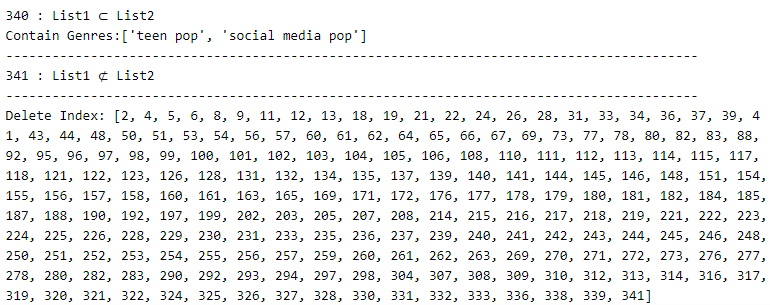
* 定義兩個list：List1為總類別清單，List2為每一位歌手的類別清單。
* 檢查歌手的類別有沒有包含在總類別中

any：歌手任何一個類別不包含在總類別中就不刪除，代表有幾個類別是要保留的

all：歌手所有類別都在總類別中就刪除

指令：check = all(item in List1 for item in List2) → 概念為陣列交集

* 若check回傳True，則代表此歌手要保留；反之，紀錄欲刪除歌手的列index，形成delete\_index。

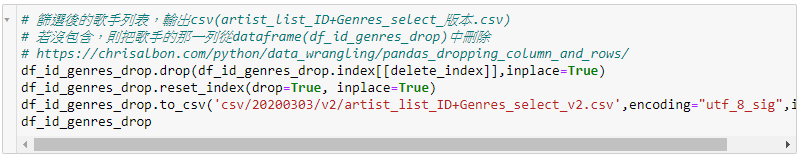




1. **剔除歌手**

* 用步驟(8)得到的delete\_index來刪除，下圖為篩選後的結果。
* 指令：df\_id\_genres\_drop.drop(df\_id\_genres\_drop.index[[delete\_index]],inplace=True)
* 輸出篩選後的歌手列表csv/20200303/v2[版本資料夾]/artist\_list\_ID+Genres\_select\_v2[版本].csv
* 教學：https://chrisalbon.com/python/data\_wrangling/pandas\_dropping\_column\_and\_rows/





1. **取得歌手的所有專輯**

* 輸入歌手id，取得歌手專輯，使用set()移除重複的專輯並使用sort()按照專輯的名稱排序。
* 將專輯的name、id、uri和href分別存放到4個list，同時放入dataframe中。
* 輸出：表格df\_albums



1. **輸出每位歌手專輯資料的csv**

* 匯入步驟(9)產生的歌手id和genres的csv (df\_id\_genres)，分別讀取欄位 'Artist' 和 'Id' 。
* 根據步驟(10)回傳的df\_albums，利用.empty檢查是否有專輯data

若有專輯data，則df\_albums.empty == False。

若沒專輯data，則df\_albums.empty == True。

* 有專輯的就輸出csv，命名格式'csv/20200303/v2/v2\_albums/df\_albums\_'+ [歌手名稱] +'.csv'。

