

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import requests
import json
import time
from datetime import datetime
import matplotlib.patches as mpatches

API_KEY = '670856810f366a1dce53a40a4d69e0df'
USER_AGENT = 'Mozilla/5.0'
USERNAME = 'R1ZEN'
```

```
In [2]: def lastfm_get(payload):
    headers = {'user-agent': USER_AGENT}
    url = 'http://ws.audioscrobbler.com/2.0/'
    payload['user'] = USERNAME
    payload['api_key'] = API_KEY
    payload['format'] = 'json'

    response = requests.get(url, headers=headers, params=payload)
    return response

# Initialize an empty dictionary as cache
artist_tags_cache = {}

def get_artist_top_tags(artist):
    # Check if the artist's tags are already in the cache
    if artist in artist_tags_cache:
        return artist_tags_cache[artist]

    # If not, fetch the tags from the API
    payload = {
        'method': 'artist.getTopTags',
        'artist': artist
    }
    response = lastfm_get(payload)
    if response.status_code != 200:
        return None
    json_response = response.json()
    if 'toptags' in json_response and 'tag' in json_response['toptags']:
        tags = [tag['name'] for tag in json_response['toptags']['tag'][:3]]
    else:
        tags = []
    time.sleep(0.5) # Delay to avoid overwhelming the API

    # Save the tags in the cache
    artist_tags_cache[artist] = ', '.join(tags)

    return artist_tags_cache[artist]

def pprint(obj):
    text = json.dumps(obj, sort_keys=True, indent=4)
    print(text)

def get_paginated_data(method, total_pages):
    page = 1
    while page <= total_pages:
        payload = {
            'method': method,
            'page': page
        }
        response = lastfm_get(payload)
        if response.status_code != 200:
            break
        json_response = response.json()
        yield json_response
        page += 1
        time.sleep(0.5) # delay between requests
```

```
In [ ]: # Get the total number of pages
response = lastfm_get({'method': 'user.getrecenttracks'})

# Extract total number of pages from the response.
# This assumes that the response has a 'recenttracks' key with an '@attr' sub-key containing 'totalPages'.
total_pages = int(response.json()['recenttracks']['@attr']['totalPages'])

# Fetch all the data pages
data = [page_data for page_data in get_paginated_data('user.getrecenttracks', total_pages)]
```

```
In [5]: # Cell 4
# Convert the nested JSON into a flat table
df = pd.json_normalize([item for sublist in [page['recenttracks']['track'] for page in data] for item in sublist])

# Drop unnecessary columns
df.drop(columns=['@attr.nowplaying', 'image', 'streamable', 'url', 'mbid', 'album.mbid', 'artist.mbid'], inplace=True)

# Extract '#text' from 'artist', 'album' and 'date' if they exist
if 'artist.#text' in df.columns:
    df['artist'] = df['artist.#text']
if 'album.#text' in df.columns:
    df['album'] = df['album.#text']
if 'date.#text' in df.columns:
    df['date'] = df['date.#text']

# Drop the original 'artist', 'album' and 'date' columns
df.drop(columns=['artist.#text', 'album.#text', 'date.#text', 'date.uts'], errors='ignore', inplace=True)

# Rename 'name' to 'track'
df.rename(columns={'name': 'track'}, inplace=True)

# Display a sample of the cleaned data
print(df.sample(5))
```

74852		track	artist \
44564		イエスタデイ	セキトオ・シゲオ
56351		kit	Julie
137057	Layla - Acoustic; Live at MTV Unplugged,	Come in	Weatherday
8253		Bray ...	Eric Clapton
		Datura	Cortex

	album	date	
74852	華麗なるエレクトーン -ザ・ワード-	28 Aug 2021, 20:40	
44564	starjump/kit	08 Jul 2022, 22:43	
56351	Come In	07 Apr 2022, 19:28	
137057	Unplugged (Deluxe Edition)	01 Apr 2020, 05:40	
8253	Cortex, Vol. 2	29 Apr 2023, 01:45	

```
In [6]: # Cell 5
# Get a list of unique artists
unique_artists = df['artist'].unique()

# Get the top tags for each unique artist
artist_tags = {artist: get_artist_top_tags(artist) for artist in unique_artists}

# Map the artist column to the genre tags
df['tags'] = df['artist'].map(artist_tags)

# Display a sample of the cleaned data with tags
print(df.sample(5))

# Save the DataFrame to a CSV file
df.to_csv('lastfm_data2.csv', index=False)
```

78861		track	artist \
68952		etoile et toi	物語シリーズ
135697		Falling for You	Weezer
76454		Eclipse	Pink Floyd
114693	Blue Monday - 2016 Remaster	Meat Grinder	Madvillain
			New Order

	album	date \	
78861	Kizumonogatari Gekihanongakushu (Original Soun...	12 Jul 2021, 04:29	
68952		Pinkerton	26 Nov 2021, 03:59
135697	The Dark Side of the Moon	08 Apr 2020, 20:30	
76454		Madvillainy	10 Aug 2021, 01:37
114693	Singles (2016 Remaster)	13 Sep 2020, 18:51	

	tags	
78861	Soundtrack, j-pop, japanese	
68952	rock, alternative, alternative rock	
135697	Progressive rock, classic rock, Psychedelic Rock	
76454	Hip-Hop, underground hip-hop, rap	
114693	new wave, electronic, post-punk	

In []: