Music-Seach Version 1 - Dokumentation

Table of contents

1 Das vollständige Python-Skript: 2

2 Erläutrungen: 3

3 Die Ausgabe des Programms im Terminal 6

Music-Search ist ein Lernprojekt, das erklärt, wie man in einem Python Skript ein Modul importiert, Benutzereingaben abfragt und eine API-Abfrage integriert.

Wir verwenden das Modul „requests" und die iTunes-API. Der Benutzer wird aufgefordert, einen Suchbegriff einzugeben, der dann in der iTunes-Datenbank gesucht wird. Künstler und Alben, die dem Suchbegriff entsprechen, werden als Ausgabe im Terminal zurückgegeben.

Voraussetzungen:

- Eine aktuelle Python-Installation auf Ihrem Computer,
- der Paketmanager pip,
- ein Editor oder vorzugsweise eine IDE.

Erstellen Sie eine virtuelle Umgebung für das Projekt.

1 Das vollständige Python-Skript:

```python
import requests

BASE_URL = "https://itunes.apple.com/search"

search_term = input("Enter a search term: ")

payloyd = {
    "term": search_term,
    "entity": "album",
    "limit": 5
}

response = requests.get(BASE_URL, params=payloyd)
response_dict =response.json()

result_count = response_dict["resultCount"]
print("The search returned ", result_count, "results.")

for result in response_dict["results"]:
    artist = result["artistName"]
    album = result["collectionName"]
    tracks = result["trackCount"]
    print("Artist:", artist)
    print("Album:", album)
    print("Track Count:", tracks)

#print(f"completed url: ", {response.url})
#print(response_json)
```

---

2 Erläutrungen:

```
1 import requests
```

Das Schlüsselwort `import` macht Code aus einem Modul zugänglich für die Ausführung in Ihrem Skript. `requests` ist ein Modul (Bibliothek) das es ermöglicht, HTTP-Requests zu senden und die Antworten vom Server, HTTP-Response, weiter zu verarbeiten.

```
1 BASE_URL = "https://itunes.apple.com/search"
```

BASE_URL ist eine statische globale Variable, die die Basis-URL der Apple iTunes Datenbank aufnimmt. Das vereinfacht später die Verwendung im Code.

```
1 search_term = input("Enter a search term: ")
```

Die Python-Funktion `input()` ermöglicht es dem Benutzer, Text einzugeben, der in der Variablen `search_term` gespeichert wird. Der Parameter für input() ist eine Eingabeaufforderung, die im Terminal angezeigt wird und den Benutzer auffordert, einen Suchbegriff einzugeben.

```
1 payloyd = {
2     "term": search_term,
3     "entity": "album",
4     "limit": 5
5 }
```

Andere entity types: - musicArtist, - musicTrack, - song

Ein Dictionary mit den für die Suchanfrage erforderlichen Parametern wird in der Variablen `payload` gespeichert. Der vom Benutzer eingegebene Suchbegriff wird in `search_term` gespeichert. Ansonsten finden Sie die möglichen Bezeichnungen – wie Entität und Begriff – in der API-Dokumentation.

„limit” ist ein Wert, den wir nur für die ersten Tests festlegen. Später können Sie ein weiteres Eingabefeld erstellen, in dem der Benutzer die Anzahl der anzuzeigenden Ergebnisse selbst festlegen kann.

https://developer.apple.com/library/archive/documentation/AudioVideo/Conceptual/iTuneSearchAPI/Searching.html#//apple_ref/doc/uid/TP40017632-CH5-SW2

```
response = requests.get(BASE_URL, params=payloyd)
```

Die Variable `response` das response-Objekt, das vom iTunes-Server zurückgegeben wird.

Die Funktion get() aus der Bibliothek `requests` sendet eine GET-Anfrage an eine Webadresse um hier Daten ITunes-API abzurufen. Sie gibt ein Response-Objekt zurück, das den Server-Status, den Inhalt (Text, JSON), Header und mehr enthält, um die abgerufene Ressource einfach weiterzuverarbeiten

```
response_dict = response.json()
```

Die `response.json()` Methode wird verwendet, um den JSON-formatierten Inhalt einer HTTP-Antwort in ein natives Python-Objekt (typischerweise ein Dictionary oder eine Liste) zu deserialisieren (umzuwandeln). Hier speichern wir das Dictionary in der Variablen response_dict.

Lässt man sich das Dictionary mit einem print-Befehl ausgeben, erhält man Folgendes:

```
response_dict: {
    'resultCount': 2,
    'results': [
        {'wrapperType': 'collection', 'collectionType': 'Album',
        'artistId': 454295032,
        'collectionId': 1533983552,
        'amgArtistId': 140996,
        'artistName': 'Hans Zimmer',
```

```
        'collectionName': 'Interstellar (Original Motion Picture
            Soundtrack) [Expanded Edition]',
        'collectionCensoredName': 'Interstellar (Original Motion Picture
            Soundtrack) [Expanded Edition]',
        'artistViewUrl': 'https://music.apple.com/us/artist/...
        ...,
        'collectionViewUrl': 'https://music.apple.com/us/album/
        ...,
        'artworkUrl60': 'https://is1-ssl.mzstatic.com/image
        ...,
        'artworkUrl100': 'https://is1-ssl.mzstatic.com/image/
        ...,
        'collectionPrice': 14.99,
        'collectionExplicitness': 'notExplicit',
        'trackCount': 30,
        'copyright': '℗ 2014 This compilation
            WaterTower Music as licensee for Warner Bros.Entertainment Inc.',
        'country': 'USA',
        'currency': 'USD',
        'releaseDate': '2014-11-18T08:00:00Z',
        'primaryGenreName': 'Soundtrack'},
        {'wrapperType': 'collection',
        'collectionType': 'Album',
        ...
        'primaryGenreName': 'Classical'
        }
    ]
}
```

Wir haben die Ausgabe für eine bessere Lesbarkeit gekürzt. Das Dictionary enthält zwei keys:

- resultCount mit dem Wert 2
- results hat als Wert eine Liste [], die selbst wieder zwei Dictionaries {} enthält.

Sie können daran erkenne, welche Schlüssel zur Verfügung stehen, um Werte abzufragen. Das tun wir dann auch in der nächsten Zeile:

```
result_count = response_dict["resultCount"]
```

Wir rufen aus dem Dictionary den Wert zum Schlüssel `resultCount` ab und speichern ihn in der Variablen `result_count`.

```
print("The search returned ", result_count, "results.")
```

Mit dem print-Befehl wird der Inhalt der Variabeln `result_count` ausgegeben.

```
for result in response_dict["results"]:
    artist = result["artistName"]
    album = result["collectionName"]
    tracks = result["trackCount"]
    print("Artist:", artist)
    print("Album:", album)
    print("Track Count:", tracks)
```

Schließlich iterieren wir mit einer for-Schleife über die Liste, die dem key `results` zugeordnet ist und speichern die Werte der einzelnen keys, die wir anzeigen möchten, jeweils in einer Variablen. Der Inhalt dieser Variablen wird in jedem Schleifendurchgang mit `print` ausgegeben.

3 Die Ausgabe des Programms im Terminal

(auf 2 Ergebnisse begrenzt)

```
Enter a search term: Forelle
The search returned  2 results.
Artist: Hans Zimmer
Album: Interstellar (Original Motion
        Picture Soundtrack) [Expanded Edition]
Track Count: 30
Artist: Various Artists
Album: Happy Classical Music
Track Count: 22
```