

2022

Dokumentation Spotify API

PROJEKTARBEIT
KEVIN AYALA

Inhalt

Projektbeschreibung.....	2
Nötige Materialien/Wissen.....	2
Arbeitsverlauf.....	2
Meilensteile.....	3
Schwierigkeiten	5
Gewonnenes Wissen	5
Verbesserungen.....	5

Projektbeschreibung

In diesem Modul dreht sich alles um API-Nutzung mit der Programmiersprache Python. Als Projekt nahm ich mir vor die API von Spotify zu erlernen und nutzen. Mein Ziel war es die beliebtesten Lieder eines Accounts zu extrahieren und als PDF weiter zu verschicken. Dies sollte per E-Mail erledigt werden.

Nötige Materialien/Wissen

- Python Verständnis
- Spotify API-Dokumentation verstehen
- Neue Entwicklungsumgebung Pycharm kennenlernen
- Vermehrt mit Modulen arbeiten

Arbeitsverlauf

Anfangen hatte ich damit die Entwicklungsumgebung Pycharm herunterzuladen. Hierbei lies ich mich in der Doku zurecht und probierte ein Hello World aus. Danach fing ich an die Spotify Dokumentation zu lesen und herauszufinden wie ich mir die API zu nutzen machen kann. Sobald der Zugriff gewährt ist, verarbeite ich die Informationen in eine PDF-Datei. In einem letzten Schritt schicke ich per Code die PDF-Datei an eine E-Mail-Adresse.

Meilensteile

Nach vielen Versuchen und Dokumentationen konnte ich die Daten mittels der Spotify API aufrufen. Hierbei musste ich darauf achten, dass ich mein Programm beim Spotify Dashboard für Entwickler anmelde. Nur so erhält man ein Token um auf die Daten von Spotify zuzugreifen zu können. Nachdem dies erledigt war, konfigurierte ich die gültige Session. Ich stellte sicher, dass mein Token gültig war und ich auf die Daten zugreifen konnte.


```
# Hier werden die Rechte und persönliche ID's vergeben
client_ID = 'ba90c35f938e41b293136e6e325c699d'
client_Secret = 'e788a1cd44e9438fbbd99bf00f923e81'
scope = 'user-top-read'
client_redirect_uri = 'http://localhost:99/callback'

# Hier nutzen wir die erstellten Rechte und ID's für eine Authentifikation damit wir einen Token erhalten.
sp = spotipy.Spotify(auth_manager=SpotipyOAuth(scope=scope, client_id = client_ID, client_secret = client_Secret,

# Hier werden die TopTracks gesucht mit bestimmten Parametern. Limit wird auf 50 gesetzt für die Top 50
results = sp.current_user_top_tracks(limit=50, offset=0, time_range='medium_term')
```

Eine grosse Hilfe hierbei war die Dokumentation von Spotify selber, welche mir sehr detailliert erklären konnte wie ich die Parameter anwenden soll und erfolgreich die Daten abrufen kann.

Get User's Top Items

 OAuth 2.0

Get the current user's top artists or tracks based on calculated affinity.

Request

✓ **GET** /me/top/{type}

type string

required

The type of entity to return. Valid values: `artists` or `tracks`

Example value: `"artists"`

✓ **Query**

limit integer

The maximum number of items to return. Default: 20. Minimum: 1. Maximum: 50.

`>= 0` `<= 50`

Default value: `20`

Example value: `10`

offset integer

The index of the first item to return. Default: 0 (the first item). Use with limit to get the next set of items.

Default value: `0`

Example value: `5`

time_range string

Over what time frame the affinities are computed. Valid values: `long_term` (calculated from several years of data and including all new data as it becomes available), `medium_term` (approximately last 6 months), `short_term` (approximately last 4 weeks). Default: `medium_term`

Default value: `"medium_term"`

Example value: `"medium_term"`

 -----

Als nächstes brauchte ich eine sehr lange Zeit meine Daten in ein PDF-Datei umzuwandeln. Hierbei gab es mehrere Versuche. Zuallererst versuchte ich JSON direkt in ein PDF umzuwandeln, dies klappte nicht und nach einiger Zeit entwickelte ich eine andere Idee. Ich versuchte als nächstes meine JSON Datei in Text umzuwandeln und von dort aus in ein PDF. Dies schlug abermals fehl. Zu guter Letzt fand ich eine Dokumentation zur Veranschaulichung von der JSON Datei. Dies verlief sehr gut und ich konnte meine JSON Datei in ein JPG Bild umwandeln. Dies konnte ich problemlos in ein PDF umwandeln.

```
#Hier werden die Daten aus den Listen in ein Bild umgewandelt beziehungsweise als Grafik geordnet und angezeigt
descending_order = all_songs['artist'].value_counts().sort_values(ascending=False).index
ax = sb.countplot(y=all_songs['artist'], order=descending_order)

sb.despine(fig=None, ax=None, top=True, right=True, left=False, trim=False)
sb.set(rc={'figure.figsize': (6, 7.2)})

ax.set_ylabel('')
ax.set_xlabel('')
ax.set_title('Songs per Artist in Top 50', fontsize=16, fontweight='heavy')
sb.set(font_scale=1.4)
ax.axes.get_xaxis().set_visible(False)
ax.set_frame_on(False)

y = all_songs['artist'].value_counts()
for i, v in enumerate(y):
    ax.text(v + 0.2, i + .16, str(v), color='black', fontweight='light', fontsize=14)

plt.savefig('top50_songs_per_artist.jpg', bbox_inches="tight")
```

In einem letzten Schritt schickte ich eine E-Mail mit dieser PDF-Datei angehängt. Dies verlief gut mit dem Modul smtplib und MIME.

```
message = MIMEMultipart()
message['From'] = sender
message['To'] = receiver
message['Subject'] = 'This email has an attachment, a pdf file'

message.attach(MIMEText(body, 'plain'))

pdfname = 'Image.pdf'
binary_pdf = open(pdfname, 'rb')

payload = MIMEBase('application', 'octate-stream', Name=pdfname)
# payload = MIMEBase('application', 'pdf', Name=pdfname)
payload.set_payload((binary_pdf).read())

# encoding the binary into base64
encoders.encode_base64(payload)

# add header with pdf name
payload.add_header('Content-Decomposition', 'attachment', filename=pdfname)
message.attach(payload)
```

Schwierigkeiten

- Abwesenheit: Dadurch, dass es mir in den letzten Monaten nicht immer gut ging, verpasste ich viel Stoff und auch Zeit, um an diesem Projekt zu arbeiten. Hierbei hätte ich die Zeit anpassen sollen und eine Notlösung vorausplanen in solchen Fällen.
- Schwierigkeit: Die SpotifyAPI war schwer zu verstehen und dann zu implementieren. Hierfür brauchte ich sehr viel Zeit. Das Gleiche galt für das PDF-Erstellen. Ich bin mir hierbei sehr sicher, dass es bestimmt auch einfacher gegangen wäre und ich die komplizierteste Route genommen hatte.

Gewonnenes Wissen

- Sicherer mit Python und Pycharm: Mir gefällt Python sehr und ich fand es sehr erfrischend eine neue Sprache kennen zu lernen und zu studieren. Die Entwicklungsumgebung ist fantastisch und bietet einiges an. Ich hatte viel Spass beim Programmieren.
- Spotify API überwältigt: Ich konnte mit Hartnäckigkeit und mit vieler Recherche mein Verständnis von Spotify so weit verbessern, sodass ich nun selbstbewusst darin bin die API zu nutzen.
- Viele neue Funktionen kennengelernt. Ich bin beim Programmieren auf einige Libraries und Module gestossen, welche das Leben erleichtern können.

Verbesserungen

Ein GUI fehlt in diesem Projekt, welche man mit den Funktionen verbinden könnte. Beispielsweise selbst bestimmen an, wenn die E-Mail gehen soll und sein eigenen Spotify Account nutzen.