LIGHTNING TALK

SPOTIFY API: OVERVIEW AND USE CASE

ALYSSA WARNICK-HESSE

A BRIEF INTRO TO SPOTIFY'S API DOCS AND SPOTIPY

FUN AND ALSO USEFUL! A SMALL REAL-LIFE EXAMPLE

API OVERVIEW

DEVELOPER.SPOTIFY.COM

- Extremely thorough setup guides

DISCOVER DOCS CONSOLE COMMUNITY DASHBOARD USE CASES

QUICK START GUIDES LIBRARIES REFERENCE

Web API Tutorial

Create a simple server-side application that accesses user related data through the Spotify Web API.

Note: By using the Spotify Tools, you accept our Developer Terms of Service.

Through the Spotify Web API, external applications retrieve Spotify content such as album data and playlists. To access *user-related data* through the Web API, an application must be authorized by the user to access that particular information.

In this tutorial we create a simple application using Node.js and JavaScript and demonstrate how to:

- Register an application with Spotify
- Authenticate a user and get authorization to access user data
- Retrieve the data from a Web API endpoint

The authorization flow we use in this tutorial is the Authorization Code Flow. This flow first gets a code from the Spotify Accounts Service, then exchanges that code for an access token. The code-to-token exchange requires a secret key, and for security is done through direct server-to-server communication.

In this example we retrieve data from the Web API `/me` endpoint, that includes information about the current user.

The complete source code of the app that will create in this tutorial is available on GitHub.

Set Up Your Account

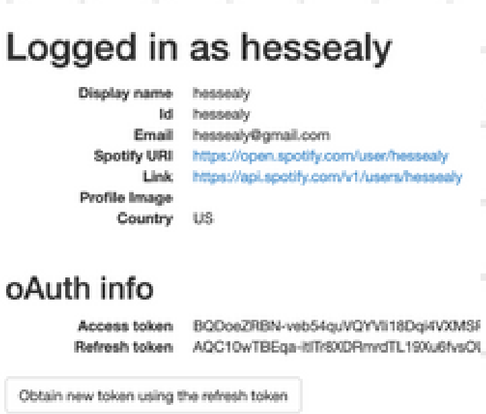
To use the Web API, start by creating a Spotify user account (Premium or Free). To do that, simply sign up at www.spotify.com.

API OVERVIEW

```
Last login: Mon Oct 25 14:30:05 on ttys001
hessealy@Alyssas-MacBook-Pro - % cd src/Spotify\ Proj
hessealy@Alyssas-MacBook-Pro Spotify Proj % cd web-api-auth-examples
hessealy@Alyssas-MacBook-Pro web-api-auth-examples % cd authorization_code
hessealy@Alyssas-MacBook-Pro authorization_code % node app.js
Listening on 8888
(node:67256) [DEP0005] DeprecationWarning: Buffer() is deprecated due to security and usability issues. Please use
(Use `node --trace-deprecation ...` to show where the warning was created)
{
  country: 'US',
  display_name: 'hessealy',
  email: 'hessealy@gmail.com',
  explicit_content: { filter_enabled: false, filter_locked: false },
  external_urls: { spotify: 'https://open.spotify.com/user/hessealy' },
  followers: { href: null, total: 0 },
  href: 'https://api.spotify.com/v1/users/hessealy',
  id: 'hessealy',
  images: [],
  product: 'premium',
  type: 'user',
  uri: 'spotify:user:hessealy'
}
{
```

Download authorization codes/files and run app.js on your local host

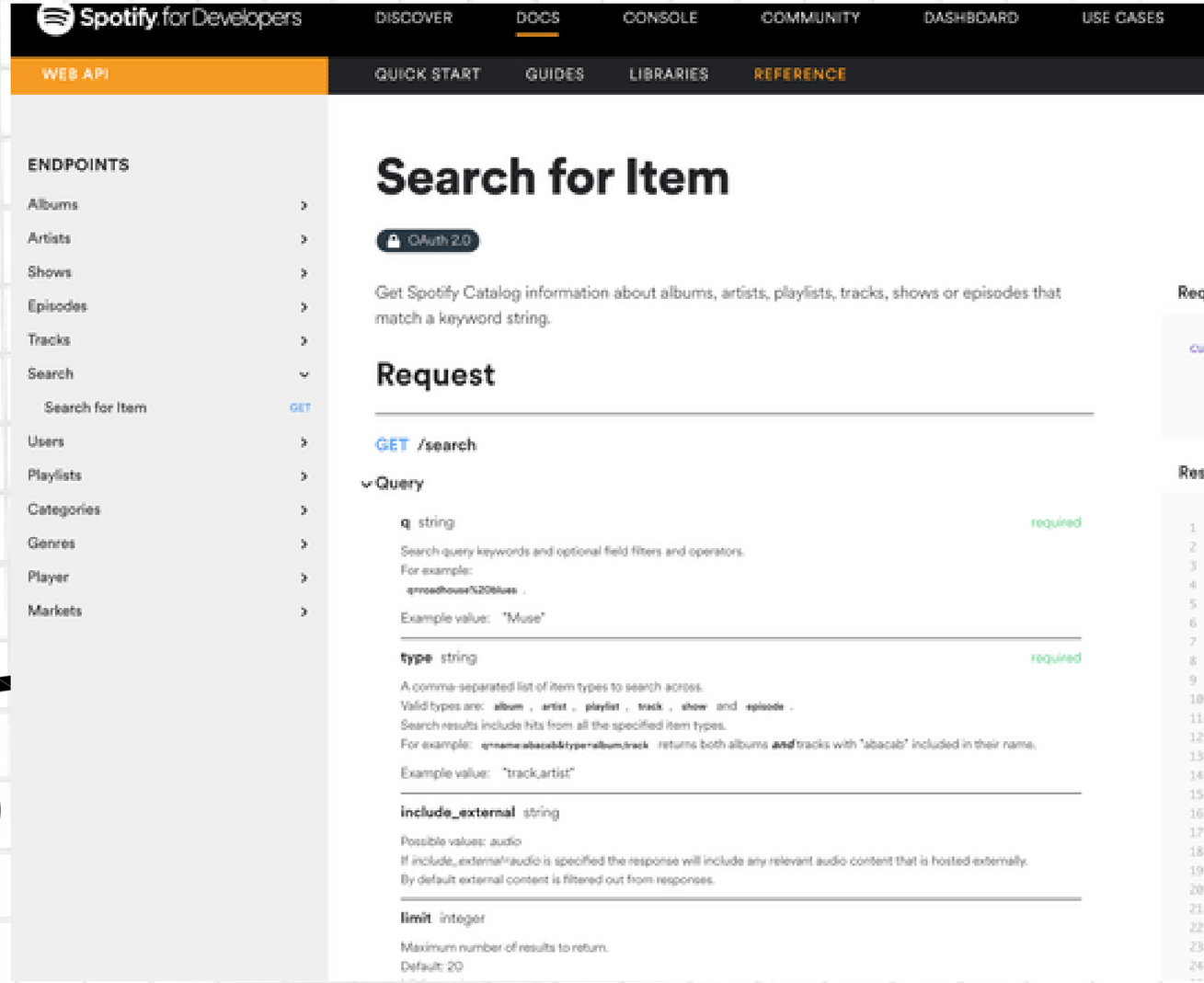
- Once logged in, you'll see your unique tokens



API OVERVIEW

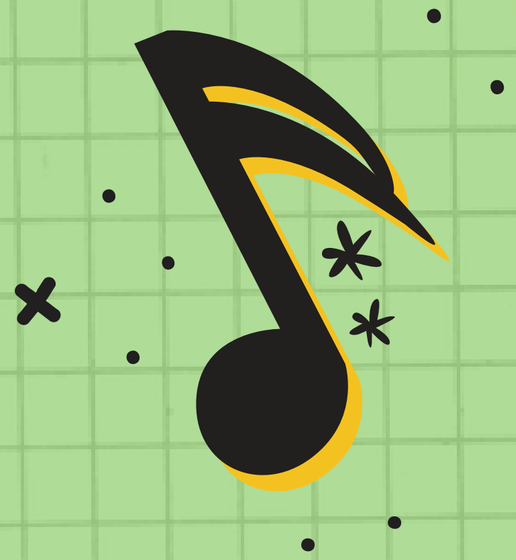
EASY TO BROWSE ENDPOINTS AND REQUESTS

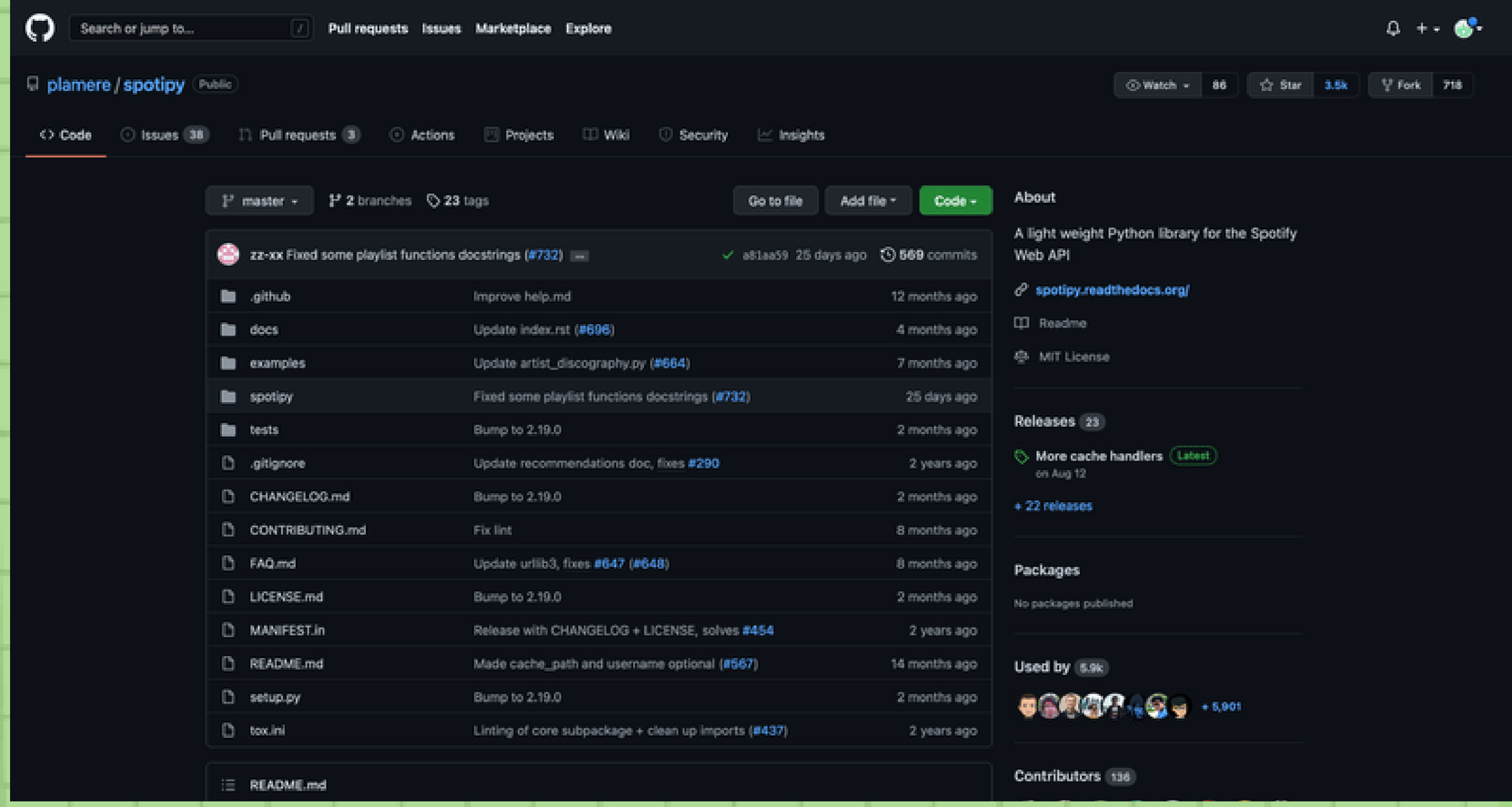
Also easy to get lost in the huge JSON data structures!



SPOTIPY!!

A HUGE LIBRARY FOR YOUR HUGE DATA

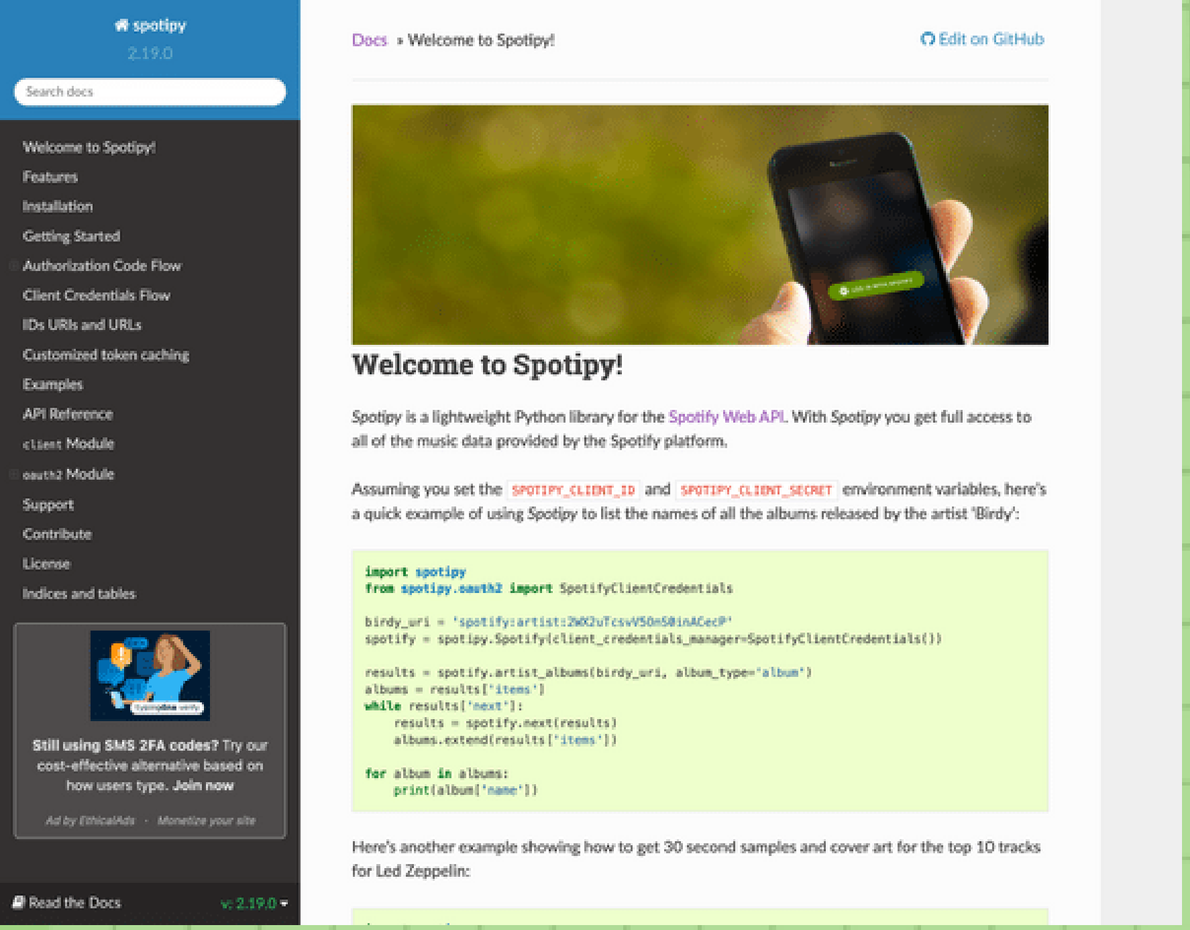




GITHUB.COM/PLAMERE/SPOTIPY

Tons of examples and walk-throughs

"client.py" has MANY pre-built functions to explore



SPOTIPY!!

spotipy.readthedocs.io/

Even more docs!

Examples and a little more reader-friendly

GITHUB.COM/PLAMERE/SPOTIPY

CLIENT.PY (OUR EXAMPLE)

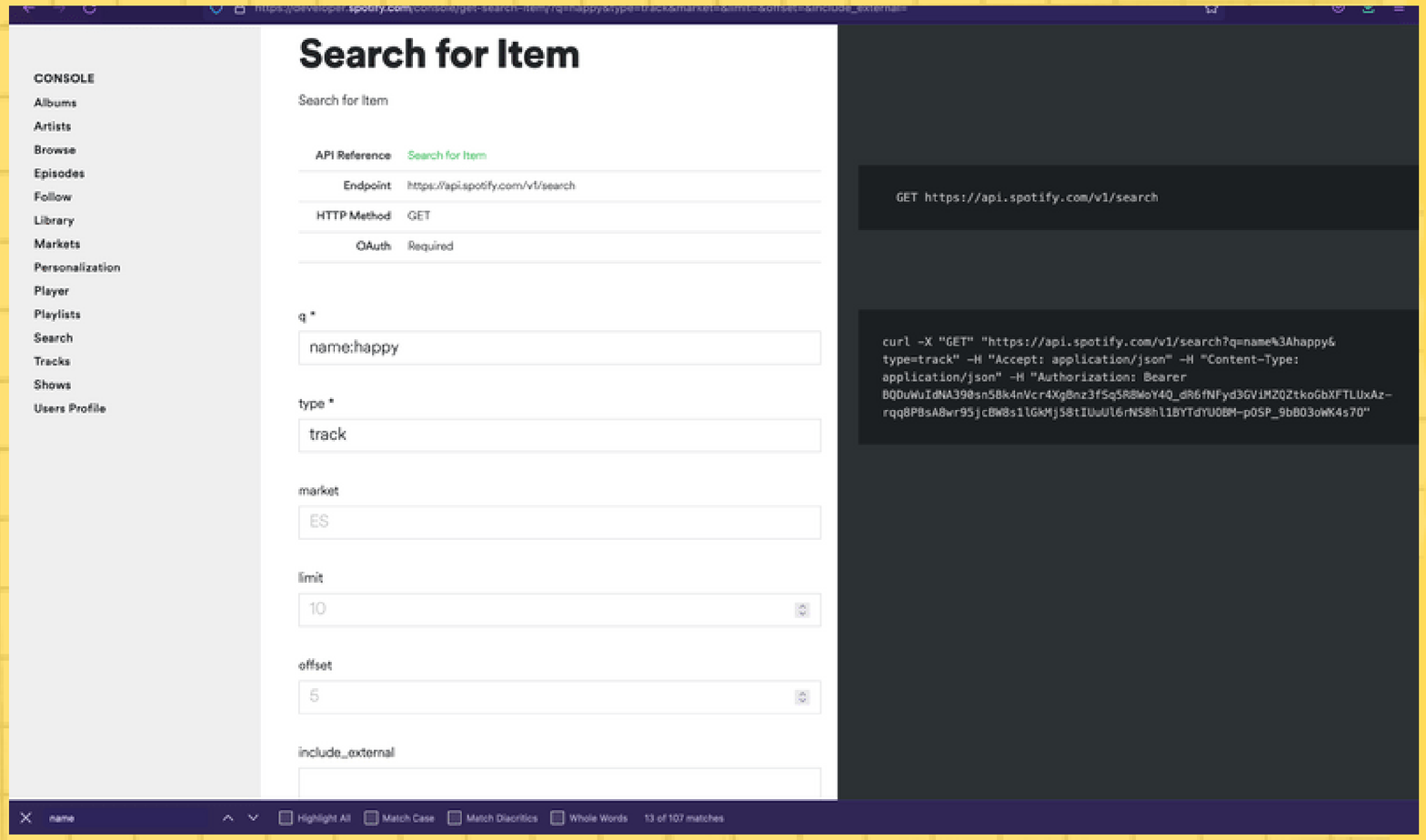
```python
    def search(self, q, limit=10, offset=0, type="track", market=None):
        """ searches for an item

            Parameters:
                - q - the search query (see how to write a query in the
                      official documentation https://developer.spotify.com/documentation/web-api/reference/search/)  # noqa
                - limit - the number of items to return (min = 1, default = 10, max = 50). The limit is applied
                          within each type, not on the total response.
                - offset - the index of the first item to return
                - type - the types of items to return. One or more of 'artist', 'album',
                         'track', 'playlist', 'show', and 'episode'.  If multiple types are desired,
                         pass in a comma separated string; e.g., 'track,album,episode'.
                - market - An ISO 3166-1 alpha-2 country code or the string
                           from_token.
        """
        return self._get(
            "search", q=q, limit=limit, offset=offset, type=type, market=market
        )
```

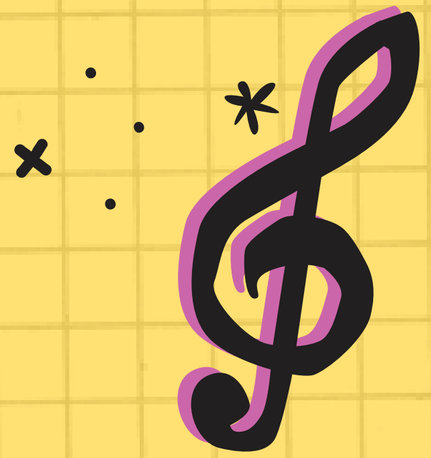
USE CASE: SPEECH THERAPY IN K-12 SCHOOLS

1. SCHOOL-BASED SLPS WORK WITH MANY STUDENTS WITH COMPLEX COMMUNICATION NEEDS
2. ONE COMMON TOOL IS TO USE A "WORD OF THE DAY/WEEK" IN SONGS OR VIDEOS TO ENCOURAGE PRACTICE
3. BUT IT'S A CHALLENGE TO FIND SONGS THAT ARE FUN, CHILD-APPROPRIATE, AND USE THE SAME WORD MANY TIMES
4. API AND SPOTIPY CAN HELP!

SEARCH FOR SONGS WITH A SPECIFIC WORD IN TITLE!

1. YOU CAN SAMPLE IN THE API CONSOLE:
2. BUT IT RETURNS A HUGE AMOUNT OF DATA
3. AND THERE'S NO EASY WAY TO CROSS-REFERENCE DATA FROM ANOTHER ENDPOINT





I used the search function included in spotipy, but wrote my own code for conditions and printing

test.py

Users > hessealy > src > Spotify Proj > env > lib > python3.8 > site-packages > spotipy > test.py > ...

```python


results = spotify.search(q="happy", type="track", limit=50)


for item in results['tracks']['items']:
    if item['explicit']:
        pass
    else:

        features = spotify.audio_features(tracks=item['id'])
        if features[0]['danceability'] > 0.75:
            print(item['name'], item['popularity'],
            item['artists'][0]['name'], features[0]['danceability'])
```

THIS RETURNS SONGS WITH "HAPPY" IN TITLE, NOT EXPLICIT, AND WITH "DANCEABILITY!"

THE RESULTS!

RUN YOUR FILE IN TERMINAL

RESULTS FOR "HAPPY"

CHANGE DANCEABILITY TO VALENCE -

NEW RESULTS FOR "HAPPY"

```
(env) hessealy@Alyssas-MacBook-Pro spotipy % python3 test.py
Happy Birthday 59 Stevie Wonder 0.779
Happy Birthday Song 57 Cocomelon 0.866
Happy Birthday Song (Trap Remix) 34 Pj Panda 0.755
Hotel Key 67 Old Dominion 0.758
Happy Now? 62 FINNEAS 0.821
Happy 48 Surface 0.792
Happy 46 Navino 0.897
(env) hessealy@Alyssas-MacBook-Pro spotipy %
```

```
(env) hessealy@Alyssas-MacBook-Pro spotipy % python3 test.py
Happy - From "Despicable Me 2" 80 Pharrell Williams 0.962
Happy Birthday to You 46 Happy Birthday TA 0.802
Happy Birthday 59 Stevie Wonder 0.96
Happy Birthday - Dance Mix 49 Happy Birthday 0.751
Happy Hour 64 Morgan Wallen 0.8
Happy Birthday/Cumpleaños Feliz 33 Dora The Explorer 0.866
Written in the Sand 68 Old Dominion 0.76
Happy Birthday Song 57 Cocomelon 0.79
Birthday - Remastered 2009 53 The Beatles 0.91
Hotel Key 67 Old Dominion 0.964
Happy Idiot 63 TV On The Radio 0.863
Happy 53 The Rolling Stones 0.965
(env) hessealy@Alyssas-MacBook-Pro spotipy %
```

WHAT DID WE LEARN?

1. DETAILED DOCS CAN BE FRIEND OR FOE
2. DAUNTING CODE FILES ARE A GREAT WAY TO PRACTICE READING CODE AND SKIMMING
3. PERSEVERANCE IS MORE IMPORTANT THAN KNOWING IT ALL

FURTHER STUDY:

- GETTING A LARGER LIST OF SONGS
- INTEGRATION WITH "GENIUS" TO SEARCH BY LYRICS
- HOW TO REMOVE INTERFERING DATA LIKE ALBUM NAME