SI 506 Week 9

Objectives

1. Write a function called **get_from_itunes** which should take 3 inputs (1 required, 2 optional):
    - a musical artist name or band name to search for (required)
    - a media type, either “song” or “musicVideo”, whose default value is “song”
    - a maximum number of results to search for, whose default value is 50

   The function should make a request to the iTunes API and return **one Python dictionary representing data from the API as a result of the search.**

2. Write a function called **get_titles** which should take 3 inputs:

- a musical artist name or band name to search for (required)
- a media type, either “song” or “musicVideo”, whose default value is “song”
- a maximum number of results to search for, whose default value is 50

(These should look familiar…)

This function should invoke the **get_from_itunes** function and use that function's return value, plus more code in the function body, to **return a list of song titles (strings).**

Materials

- The iTunes search API. Documentation can be seen here: https://www.apple.com/itunes/affiliates/resources/documentation/itunes-store-web-service-search-api.html
- **Base url:** https://itunes.apple.com/search
- **Params you’ll use:**
    - **term:** this is any term to search for – a band name, singer, music artist, etc.
    - **entity:** this parameter can be one of two values, either **song** or **musicVideo**.
    - **limit:** The limit parameter specifies how many results the API should return for your request. The value can be any integer from 1-200, and the default from the API is 50.
- You'll need to open your own .py files and save them to complete this!

Step 1: Testing the API

Use the requests module to make a request to the iTunes API.

At this point, you can just hard-code the parameters for **term** and **entity** (choose your favorite artist!). If you can’t think of which parameters to use, try “The Beatles” and “song.”

Ways you can test the request

- Print the response code (`<response object>.status_code`). If the request was successful, the code will be `200`.
- Print the response text (`<response object>.text`). If your request worked, you should see a lot of data, and it will probably be hard to understand (since it's not formatted). But if the text contains data related to the parameters you chose, your request is working.

Step 2: Understanding the Data

Now that you've successfully made a request, it's time to start understanding the data you're receiving. Here are some methods you can use to understand the data (choose whichever you like):

- **Look at the sample response provided by the iTunes API Documentation.**
  The iTunes API (like all good APIs) provides helpful documentation with examples of response data:

  https://affiliate.itunes.apple.com/resources/documentation/itunes-store-web-service-search-api/#understand

- **Print the response text in JSON with indentation.**
  To do this, you can use the json.dumps method with the optional `indent` parameter:

  ```
  print(json.dumps(<your response json object>, indent=2))
  ```

  Hint: You can use the `limit` parameter to limit the API response to one song, which makes it a little less overwhelming to see the structure of the data.

- **Paste the response text in JSONEditorOnline.**
  You can paste your response text in the left panel at http://jsoneditoronline.org/ , and use the tree structure in the right panel to navigate through the nested data.

  If you're having trouble copying and pasting from the command prompt:
  - Save the result of json.dumps in a variable
  - Open a file for writing
  - Write the string result from json.dumps to the file
  - Close the file
  - Open up that file in a text editor and copy the entire contents of the file, and paste THAT into jsoneditoronline.org
  - a maximum number of results to search for, whose default value is 50

Step 3: Defining the first function

1. Define a function **get_from_itunes** which takes 3 inputs
    a. a musical artist name or band name to search for
    b. a media type, either ‘song’ or ‘musicVideo’ (default value is “song”)
2. In this function, you will do the following:
    a. Write code to make a request to the iTunes API with the 2 inputs above.
    b. Process the JSON-formatted text data in the response so that the function will return an index-able Python object (list or dictionary) with data from the iTunes API

Step 4: Calling the Function

1. Call the function **get_from_itunes** and test out that it returns what you want it to return (a Python dictionary)

Step 5: Defining the second function

1. Define a function **get_titles** which takes 3 inputs (see above)
2. Invoke the **get_from_itunes** function inside this function definition
3. Use its return value to iterate through the correct data in the return value from **get_from_itunes**, and extract the song or musicVideo titles.

    *Note that the key for this field is not called ‘title,’ but it <u>will be the same</u> whether the entity param is song or musicVideo.

Sample Output:

```
['Here Comes the Sun', 'Let It Be', 'Come Together', 'Hey Jude', 'In My Life',
'Blackbird', 'Twist and Shout', 'Yesterday', 'Something', 'While My Guitar
Gently Weeps', 'Ob-La-Di, Ob-La-Da', 'Revolution', 'With a Little Help From My
Friends', 'Lucy In the Sky With Diamonds', "When I'm Sixty-Four", 'Oh!
Darling', 'I Saw Her Standing There', 'Eleanor Rigby', 'Golden Slumbers',
'Strawberry Fields Forever', 'A Day In the Life', 'Birthday', "Sgt. Pepper's
Lonely Hearts Club Band", "Don't Let Me Down", 'Help!', "I Want You (She's So
Heavy)", "Octopus's Garden", 'Carry That Weight', 'Back In the U.S.S.R.', 'Dear
Prudence', 'Because', "Maxwell's Silver Hammer", ‘Rocky Raccoon', 'She Came In
Through the Bathroom Window', 'You Never Give Me Your Money', 'The End', 'I Am
the Walrus', 'Sun King', 'Helter Skelter', ‘'Mean Mr. Mustard', 'Norwegian Wood
(This Bird Has Flown)', 'Polythene Pam', u'I Will', 'Her Majesty', 'Happiness
Is a Warm Gun', 'Let It Be', 'All You Need Is Love', 'Getting Better', 'I Want
to Hold Your Hand', 'Lovely Rita']
```

Challenges if you are done:

- Take a look at the API documentation. How could you change your code so that you could also search for the **album** entity type?
- What if you wanted to make a CSV of information about songs? What information would you want to capture? Why? What code would you write to do that?