Print to PDF

Song Genius API

Contents

In this lesson, we're going to use the Genius API to access data about Missy Elliott songs.

API Keys

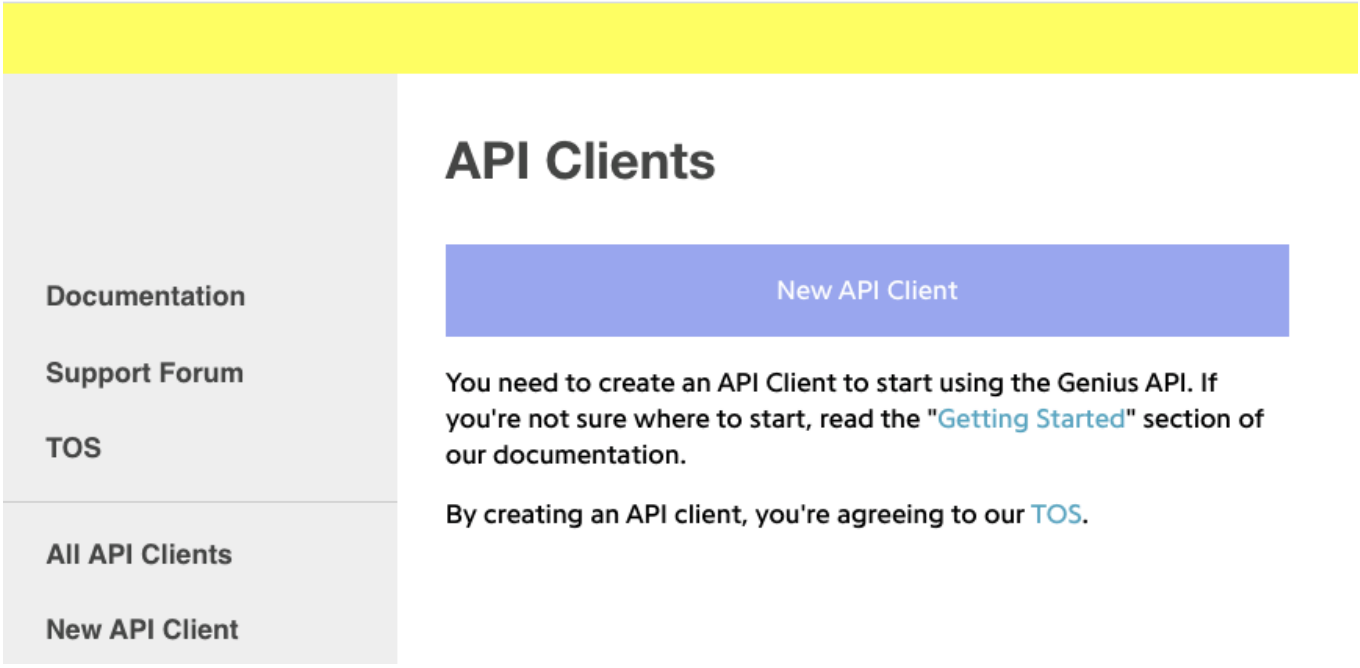
To use the Genius API, you need a special API key, specifically a "Client Access Token", which is kind of like a password. Many APIs require authentication keys to gain access to them. To get your necessary Genius API keys, you need to navigate to the following URL: https://genius.com/api-clients.

Skip to main content

You'll be prompted to sign up for a Genius account, which is required to gain API access. Signing up for a Genius account is free and easy. You just need a Genius nickname (which must be one word), an email address, and a password.

Once you're signed in, you should be taken to https://genius.com/api-clients, where you need to click the button that says "New API Client."

Skip to main content



After clicking "New API Client," you'll be prompted to fill out a short form about the "App" that you need the Genius API for. You only need to fill out "App Name" and "App Website URL."

It doesn't really matter what you type in. You can simply put "Song Lyrics Project" for the "App Name" and the URL for our course website "https://melaniewalsh.github.io/Intro-Cultural-Analytics/" for the "App Website URL."

When you click "Save," you'll be given a series of API Keys: a "Client ID" and a "Client Secret." To generate your "Client Access Token," which is the API key that we'll be using in this notebook, you need to click "Generate Access Token".

Finally, copy and paste your "Client Access Token" into the quotation marks below, and run the cell to save your variable

```
client_access_token = "INSERT YOUR CLIENT ACCESS TOKEN IN THESE QUOTATION MARKS"
```

Protecting Your API Key

For this lesson, if you just copy and paste your Genius API key into your Jupyter notebook, everything should be fine. But that's actually not the best way of storing your API keys. If you published this notebook to GitHub, for example, other people might be able to read and use/steal your API key.

Skip to main content

For this reason, it's best practice to keep your API keys away from your code, such as in another file. For example, I made a new Python file called "api_key.py" that contains just one variable `your_client_access_token = "MY API KEY"`, and I can import this variable into my notebook with `import api_key`.

```
import api_key
```

By importing this Python file/module, I get access to the variable `your_client_access_token` without ever explicitly typing my secret API token in this notebook. If I wanted to publish this notebook to GitHub, then I could ignore or leave out the "api_key.py" file that actually contains my Client Access Token.

```
api_key.your_client_access_token
```

```
'INSERT YOUR API KEY HERE'
```

Attention

You should only un-comment and run the cell below if you are importing your API key from the Python file "api_key.py". If you have already set your `client_acces_token` in this notebook, you can skip the cell below.

```
#client_access_token = api_key.your_client_access_token
```

Making an API Request

Making an API request looks a lot like typing a specially-formatted URL. But instead of getting a rendered HTML web page in return, you get some data in return.

There are a few different ways that we can query the Genius API, all of which are discussed in the Genius API documentation. The way we're going to cover in this lesson is the basic search, which allows you to get a bunch of Genius data about any artist or songs that you search for:

Skip to main content

Sticking with our Missy Elliott theme/obsession, we're going to search for Genius data about Missy Elliott.

First we're going to assign the string "Missy Elliott" to the variable `search_term`. Then we're going to make an f-string URL that contains the variables `search_term` and `client_access_token`.

```
search_term = "Missy Elliott"
```

```
genius_search_url = f"http://api.genius.com/search?q={search_term}&access_token={clien
```

This URL is basically all we need to make a Genius API request. Want proof? Run the cell below and print this URL, then copy and paste it into a new tab in your web browser.

```
print(genius_search_url)
```

```
http://api.genius.com/search?q=Missy Elliott&access_token=YOUR-API-KEY
```

It doesn't look pretty, but that's a bunch of Genius data about Missy Elliott!

We can programmatically do the same thing by again using the Python library `requests` with this URL. Instead of getting the `.text` of the response, as we did before, we're going to use `.json()`.

JSON is a data format that is commonly used by APIs. JSON data can be nested and contains key/value pairs, much like a Python dictionary.

```
import requests
```

```
response = requests.get(genius_search_url)
json_data = response.json()
```

The JSON data that we get from our Missy Elliott API query looks something like this:

Skip to main content

```
{'meta': {'status': 200},
 'response': {'hits': [{'highlights': [],
    'index': 'song',
    'type': 'song',
    'result': {'annotation_count': 32,
     'api_path': '/songs/4176',
     'full_title': 'Work It by\xa0Missy\xa0Elliott',
     'header_image_thumbnail_url': 'https://images.genius.com/27c1fbfef17041b435302af
     'header_image_url': 'https://images.genius.com/27c1fbfef17041b435302af288cfa0c6
     'id': 4176,
     'lyrics_owner_id': 6654,
     'lyrics_state': 'complete',
     'path': '/Missy-elliott-work-it-lyrics',
     'pyongs_count': 39,
     'song_art_image_thumbnail_url': 'https://images.genius.com/27c1fbfef17041b435302
     'song_art_image_url': 'https://images.genius.com/27c1fbfef17041b435302af288cfa0c
     'stats': {'unreviewed_annotations': 2,
      'hot': False,
      'pageviews': 1205121},
     'title': 'Work It',
```

We can index this data (again, like a Python dictionary) and look at the first "hit" about Missy Elliott from Genius.com.

```
json_data['response']['hits'][0]
```

```
{'highlights': [],
 'index': 'song',
 'type': 'song',
 'result': {'annotation_count': 32,
  'api_path': '/songs/4176',
  'full_title': 'Work It by\xa0Missy\xa0Elliott',
  'header_image_thumbnail_url': 'https://images.genius.com/27c1fbfef17041b435302af288cf
  'header_image_url': 'https://images.genius.com/27c1fbfef17041b435302af288cfa0c6.482x5
  'id': 4176,
  'lyrics_owner_id': 6654,
  'lyrics_state': 'complete',
  'path': '/Missy-elliott-work-it-lyrics',
  'pyongs_count': 39,
  'song_art_image_thumbnail_url': 'https://images.genius.com/27c1fbfef17041b435302af288
  'song_art_image_url': 'https://images.genius.com/27c1fbfef17041b435302af288cfa0c6.482
  'stats': {'unreviewed_annotations': 2, 'hot': False, 'pageviews': 1205121},
  'title': 'Work It',
  'title_with_featured': 'Work It',
  'url': 'https://genius.com/Missy-elliott-work-it-lyrics',
  'song_art_primary_color': '#d63220',
  'song_art_secondary_color': '#512913',
  'song_art_text_color': '#fff',
```

Skip to main content

```
    'id': 1529,
    'image_url': 'https://images.genius.com/085828b7d79bf8cf068b1557ca7a5e4c.1000x1000x1
    'is_meme_verified': False,
    'is_verified': False,
    'name': 'Missy Elliott',
    'url': 'https://genius.com/artists/Missy-elliott'}}}
```

We can tell that this data describes the song "Work It" and contains other information about the song, such as its number of Genius annotations, its number of web page views, and links to images of its album cover.

Looping Through JSON Data

Get Song Titles

```
for song in json_data['response']['hits']:
    print(song['result']['full_title'])
```

```
Work It by Missy Elliott
WTF (Where They From) by Missy Elliott (Ft. Pharrell Williams)
Get Ur Freak On by Missy Elliott
Gossip Folks by Missy Elliott (Ft. Ludacris)
I'm Better by Missy Elliott (Ft. Cainon Lamb)
The Rain (Supa Dupa Fly) by Missy Elliott
Lose Control by Missy Elliott (Ft. Ciara & Fatman Scoop)
One Minute Man (Amended Version) by Missy Elliott (Ft. Ludacris & Trina)
This Is Me (The Reimagined Remix) by Keala Settle, Kesha & Missy Elliott
Bomb Intro/Pass That Dutch by Missy Elliott
```

Get Song Tiles and Page View Counts

```
for song in json_data['response']['hits']:
    print(song['result']['full_title'], song['result']['stats']['pageviews'])
```

```
Work It by Missy Elliott 1205121
WTF (Where They From) by Missy Elliott (Ft. Pharrell Williams) 289808
Get Ur Freak On by Missy Elliott 165394
```

Skip to main content

```
The Rain (Supa Dupa Fly) by Missy Elliott 103348
Lose Control by Missy Elliott (Ft. Ciara & Fatman Scoop) 93331
One Minute Man (Amended Version) by Missy Elliott (Ft. Ludacris & Trina) 74646
This Is Me (The Reimagined Remix) by Keala Settle, Kesha & Missy Elliott 49366
Bomb Intro/Pass That Dutch by Missy Elliott 48848
```

Transform Song Titles and Page View Counts into a DataFrame

We can loop through this data, append it into a list, and then transform that list into a Pandas dataframe by calling `pd.DataFrame()`

```
import pandas as pd
```

> 🐼 Pandas Review
>
> Do you need a refresher or introduction to the Python data analysis library Pandas? Be sure to check out Pandas Basics (1-3) in this textbook!

```
missy_songs = []
for song in json_data['response']['hits']:
    missy_songs.append([song['result']['full_title'], song['result']['stats']['pagevie

#Make a Pandas dataframe from a list
missy_df = pd.DataFrame(missy_songs)
missy_df.columns = ['song_title', 'page_views']
missy_df
```

Skip to main content

	song_title	page_views
0	Work It by Missy Elliott	1205121
1	WTF (Where They From) by Missy Elliott (Ft. Ph...	289808
2	Get Ur Freak On by Missy Elliott	165394
3	Gossip Folks by Missy Elliott (Ft. Ludacris)	111483
4	I'm Better by Missy Elliott (Ft. Cainon Lamb)	107273
5	The Rain (Supa Dupa Fly) by Missy Elliott	103348
6	Lose Control by Missy Elliott (Ft. Ciara & Fat...	93331
7	One Minute Man (Amended Version) by Missy Elli...	74646
8	This Is Me (The Reimagined Remix) by Keala Set...	49366
9	Bomb Intro/Pass That Dutch by Missy Elliott	48848

Transform Song Titles, Page View Counts, & Album Covers into a DataFrame

Just for fun, we can do the same thing but also add links to images of Missy Elliott's album art—and we can actually display those images, too!

To display images in a Pandas dataframe, you need to run `from IPython.core.display import HTML` and make the function `get_image_html()`. We're going to take the image URLs and make them into HTML objects.

```
from IPython.core.display import HTML
```

```
def get_image_html(link):
    image_html = f"<img src='{link}' width='500px'>"
    return image_html
```

```
missy_songs = []
for song in json_data['response']['hits']:
    missy_songs.append([song['result']['full_title'], song['result']['stats']['pageview
```

Skip to main content

```
missy_df.columns = ['song_title', 'page_views','album_cover_url']

#Use the function get_image_html()
missy_df['album_cover'] = missy_df['album_cover_url'].apply(get_image_html)
missy_df
```

Skip to main content