
Spotichart

Release 1.0.0

Jun 19, 2020

Contents

1	spotichart package	3
1.1	Subpackages	3
1.1.1	spotichart.language package	3
1.1.2	spotichart.lyrics package	3
1.1.3	spotichart.spotipy package	5
1.2	Module contents	7
1.3	Domain Model	7
1.4	Model View	7
2	Indices and tables	9
3	Spotichart	11
3.1	Introduction	11
3.2	Documentation	11
3.3	Installation	11
3.4	Requirements	11
3.5	Synopsis	12
3.5.1	Usage	12
3.5.2	The DataFrame	12
	Python Module Index	15
	Index	17

Contents:

1.1 Subpackages

1.1.1 spotichart.language package

Submodules

spotichart.language.main module

language detector main module

`spotichart.language.main.detect_language(text)`

Detect the language of a given text

Parameters `text` (*str*) – The text to identify the language

Returns Language code identified

Return type `str`

Module contents

language detector package.

1.1.2 spotichart.lyrics package

Submodules

spotichart.lyrics.lyrics_scraper module

spotichart.lyrics.lyrics_scraper module.

`spotichart.lyrics.lyrics_scraper.scrap_lyrics` (*lyrics_url*, *headers=False*,
ad_libs=False)

Download the lyrics from a resource located on Genius, by web scraping.

Parameters

- **lyrics_url** (*str*) – Link to the Genius lyrics
- **headers** (*bool*, *optional*) – Whether keep section headers or not. Section headers explained on: <https://genius.com/9250687>, defaults to False
- **ad_libs** (*bool*, *optional*) – Keep the ad-libs sound effects (surrounded by parenthesis). Explained on <https://genius.com/9257397>, defaults to False

Returns Lyrics found

Return type str

spotichart.lyrics.main module

spotichart.lyrics.track_features entry module.

`spotichart.lyrics.main.get_lyrics` (*access_token*, *track_id*, *track_name*, *artist*)

Get the Lyrics for an individual track

Parameters

- **access_token** (*str*) – Genius API Access Token
- **track_id** (*str*) – Spotify Track Id, to identify different tracks
- **track_name** (*str*) – Track Name to search
- **artist** (*str*) – Track's Artist or Performer

Returns Dictionary with the song Lyrics, Genius ID and Language identified

Return type dict

`spotichart.lyrics.main.get_lyrics_from_chart` (*access_token*, *chart*, *sleep=1*)

Get track lyrics from a DataFrame with 'Track Id', 'Track Name' and 'Artist' columns

Parameters

- **access_token** (*str*) – Genius API Access Token
- **chart** (*pandas.DataFrame*) – Pandas DataFrame to know Artist and Track Name
- **sleep** (*int*, *optional*) – Sleep timer to rest the scraper, defaults to 1

Returns Dataframe with Lyrics and Language identified

Return type pandas.DataFrame

spotichart.lyrics.track_features module

spotichart.lyrics.track_features module.

`spotichart.lyrics.track_features.request_song_info` (*access_token*, *track_name*, *artist*)

Search the track's metadata in Genius

Parameters

- **access_token** (*str*) – Genius API access token

- **track_name** (*str*) – Track Name
- **artist** (*str*) – Track’s Artist or Performer

Returns Genius API response

Return type json

`spotichart.lyrics.track_features.search_song(access_token, track_name, artist)`
 Locate the song’s lyrics in Genius, to know its url

Parameters

- **access_token** (*str*) – Genius API Access Token
- **track_name** (*str*) – Track Name
- **artist** (*str*) – Track’s Artist or Performer

Raises **ValueError** – Error on response

Returns Track’s lyrics and id on Genius

Return type str

Module contents

lyrics package.

1.1.3 spotichart.spotipy package

Submodules

spotichart.spotipy.audio_features module

spotichart.spotipy.audio_features module.

`spotichart.spotipy.audio_features.get_audio_features(access_token, track_id)`
 Function to fetch the audio features of a song

Parameters

- **access_token** (*str*) – Spotify Web API Access Token
- **track_id** (*str*) – Spotify Track identifier

Raises

- **ValueError** – Spotify Request Error
- **ValueError** – Http Request Error

Returns Track’s audio features

Return type dict

spotichart.spotipy.main module

spotichart.spotipy.main entry point module.

`spotichart.spotipy.main.generate_top_chart` (*access_token*, *start*, *end=None*, *region='en'*,
chart='top200', *sleep=1*)

Function to fetch the top chart for a given date, and request their audio features

Parameters

- **access_token** (*str*) – Spotify Web API Access token
- **start** (*Date*) – Starting point for the scraper to get the top chart
- **end** (*Date*, *optional*) – Interval for multi-chart table, defaults to None
- **region** (*str*, *optional*) – Spotify Top 50 region code, defaults to 'en'
- **chart** (*str*, *optional*) – Spotify chart to get the data from, either top200 or viral, defaults to 'top200'
- **sleep** (*int*, *optional*) – Sleep time for the scraper to rest, defaults to 1

Returns Dataframe that stores the chart data, and the audio features for each track

Return type pandas.DataFrame

spotichart.spotipy.top_charts module

spotichart.spotipy.top_charts module. Based upon the repo by fbkarsdorp Located on <https://github.com/fbkarsdorp/spotify-chart>

`spotichart.spotipy.top_charts.get_chart` (*date*, *region='en'*, *chart='top200'*)

Download an individual chart

Parameters

- **date** (*Date*) – Specific date for a Top Chart
- **region** (*str*, *optional*) – Spotify Top 50 region code, defaults to 'en'
- **chart** (*str*, *optional*) – Spotify chart to get the data from, either top200 or viral, defaults to 'top200'

Raises **ValueError** – Unavailable data requested

Returns Top 50 Chart

Return type pandas.DataFrame

`spotichart.spotipy.top_charts.get_charts` (*start*, *end=None*, *region='global'*,
chart='top200', *sleep=1*)

Fetch multiple Charts

Parameters

- **start** (*Date*) – Starting date to download the chart
- **end** (*Date*, *optional*) – End date for an interval of top charts, defaults to None
- **region** (*str*, *optional*) – Spotify Top 50 region code, defaults to 'global'
- **chart** (*str*, *optional*) – Spotify chart to get the data from, either top200 or viral, defaults to 'top200'
- **sleep** (*int*, *optional*) – Sleep time for the scraper to rest, defaults to 1

Raises **ValueError** – Invalid date interval format

Returns Chart with the Top 50 basic data

Return type `pandas.DataFrame`

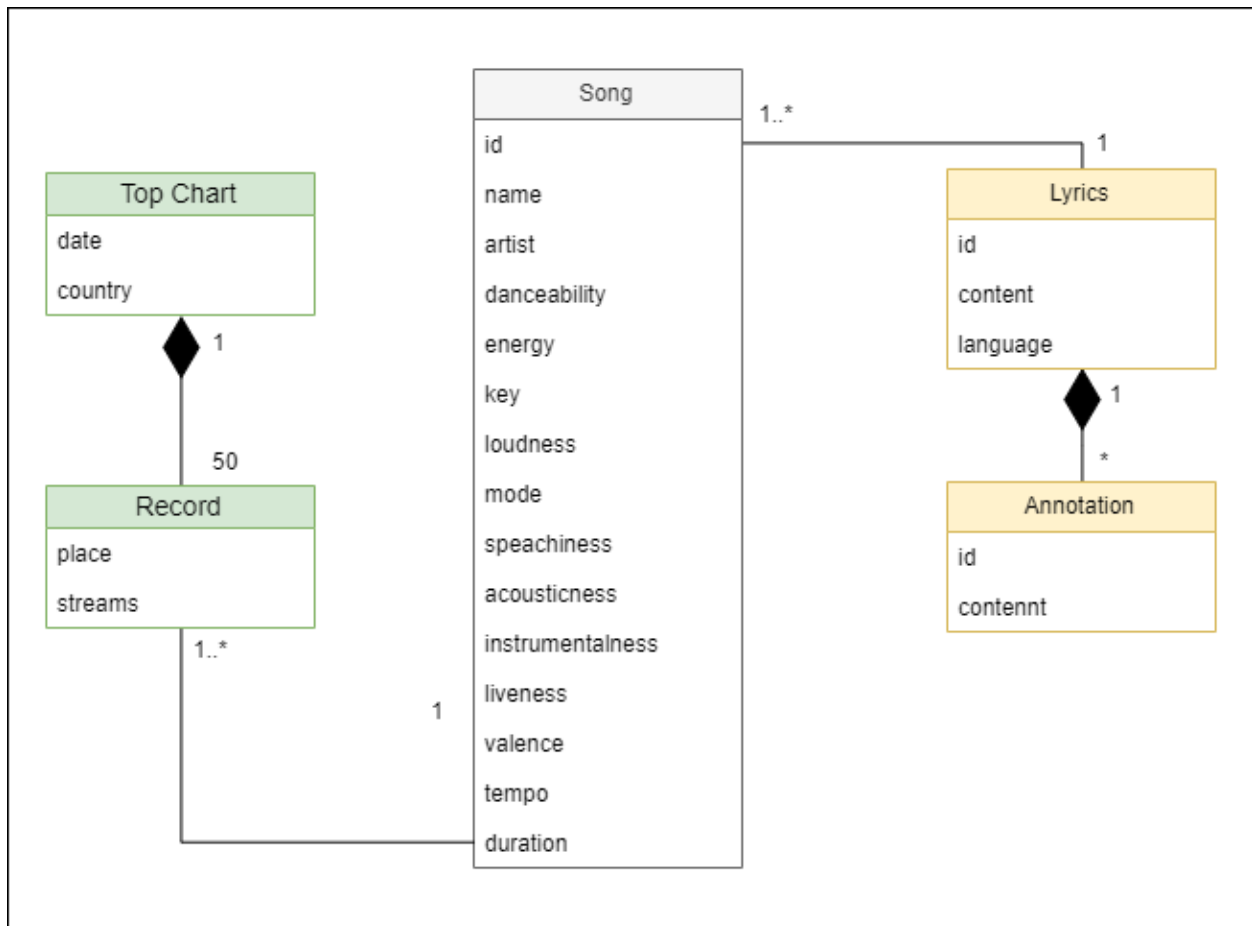
Module contents

spotipy package.

1.2 Module contents

spotichart package.

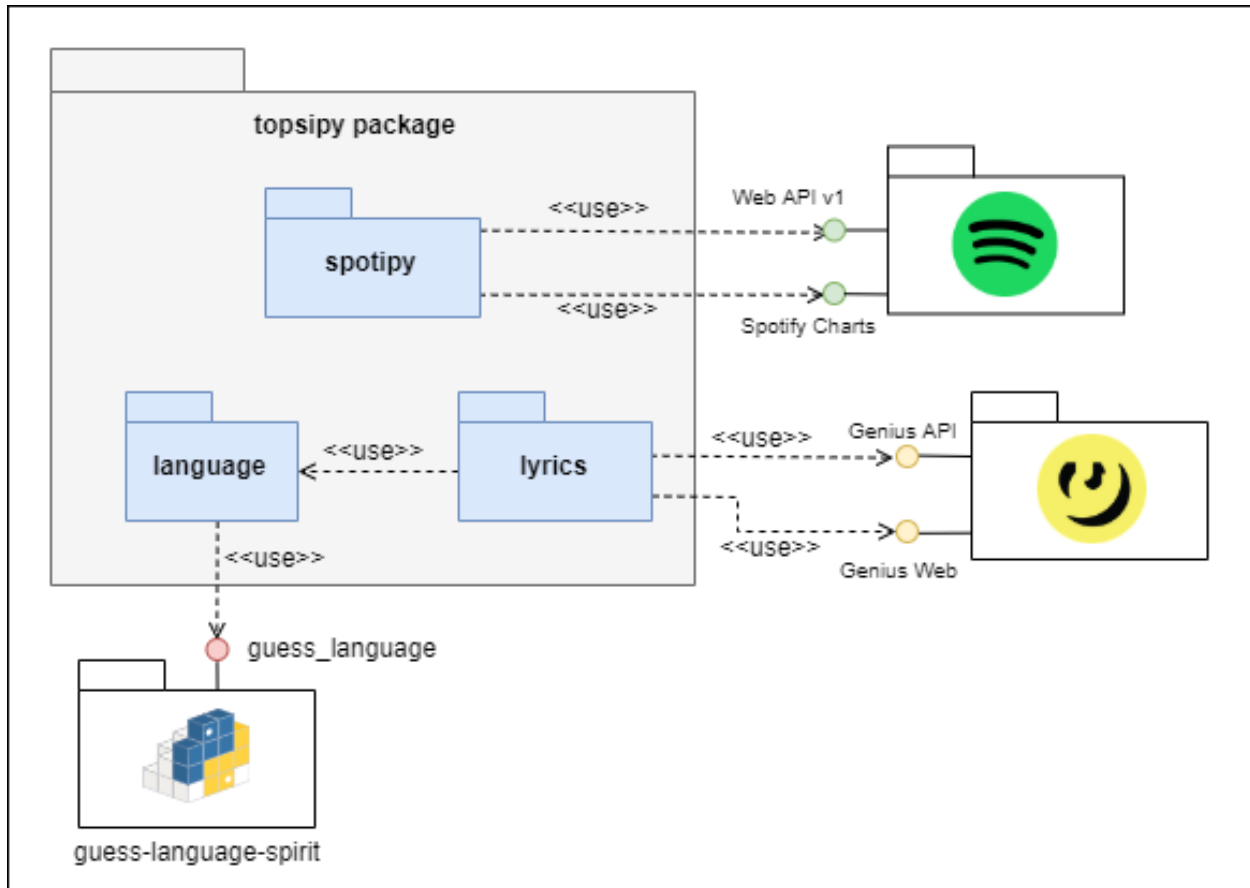
1.3 Domain Model



Attributes involved in the scope of this package. Provided from [Spotify Top Charts](#), [Spotify Web API](#), [Genius API](#), and Genius itself (by web scraping the lyrics).

1.4 Model View

Therefore, the providers and the inner modules would look this way, using the `Model View with Use Style`.



Additionally, the `guess-language-spirit` package is used to detect the lyrics language. The `language` package and its dependency are structured that way, so it is easier to switch the language-detecting provider.

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`

Collector Module for Spotify National Trending Analysis

3.1 Introduction

The Spotichart module makes it easy for data scientist and programmers get the features from the trending songs on Spotify. You can define period of time and a region and get the main characteristics of the top songs.

3.2 Documentation

The oficial documentations is available at: [Read The Docs](#)

3.3 Installation

TODO: Not Yet Published

```
$ pip install spotichart
```

3.4 Requirements

- **Python** `>= 3.6`
- **Spotify Web API Access Token**, you can request yours [here](#) and click on GET TOKEN. Then copy the token on the OAuth Token field.

- (Optional) **Genius Web API Access Token.** From [the official docs page](#) you can just select Authenticate with the Docs App To Try, and copy the Authorization Bearer provided after logging in.

3.5 Synopsis

3.5.1 Usage

Just to get the audio features, given a date (or period) and a region

```
import spotichart

spotify_token = 'YOUR-ACCESS-TOKEN-FROM-THE-WEB-API'

chart = spotichart.generate_top_chart(spotify_token, start='2019-01-01', end='2019-10-13', region='mx')
```

To additionally retrieve each song's lyrics, Genius ID and auto-detect the language, you can do as well:

```
import spotichart

spotify_token = 'YOUR-SPOTIFY-ACCESS-TOKEN-FROM-THE-WEB-API'
genius_token = 'YOUR-GENIUS-ACCESS-TOKEN-FROM-THE-WEB-API'

chart = spotichart.generate_top_chart(spotify_token, start='2019-01-01',
                                     end='2019-10-13', region='mx', sleep=0.5)

chart_with_lyrics = spotichart.get_lyrics_from_chart(genius_token, chart, sleep=0.1)
```

Note: Since these functions imply web requests to get the data, the `sleep` parameter is meant to make the algorithm rest and avoid the server to refuse the requests. By default `sleep` is set to 1 second.

3.5.2 The DataFrame

A `pandas.DataFrame` will be generated with the data of interest:

```
>>> chart
   Position  Streams  ...  speechiness  tempo  time_signature  valence  Track Name  Artist
0          1    737894  ...    0.0524    126.899             4    0.761    Calma - Remix  Pedro Capó
1          2    415066  ...    0.3360    171.993             4    0.720    Adan y Eva    Paulo Londra
2          3    409061  ...    0.2290    95.948             4    0.591    Taki Taki (with Selena Gomez, Ozuna & Cardi B)  DJ Snake
3          4    377855  ...    0.0621    97.062             4    0.158    MIA (feat. Drake)    Bad Bunny
4          5    346975  ...    0.0297    143.851             3    0.920    A Través Del Vaso  Grupo Arranke
...        ...    ...    ...    ...    ...    ...    ...    ...    ...
14295     46    141397  ...    0.0593    93.989             4    0.656    Con Calma    Daddy Yankee
14296     47    139350  ...    0.0371    112.548             4    0.844    La Escuela No Me Gustó  Adriel Favela
```

(continues on next page)

(continued from previous page)

14297	48			De Los Besos Que Te Di	Christian Nodal	└
↪139294	...	0.0422	195.593	4	0.709	
14298	49			Pa Mí - Remix	Dalex	└
↪137812	...	0.2200	170.018	4	0.727	
14299	50			Circles	Post Malone	└
↪131109	...	0.0395	120.042	4	0.5	
[14300 rows x 20 columns]						

S

- `spotichart`, 7
- `spotichart.language`, 3
 - `spotichart.language.main`, 3
- `spotichart.lyrics.lyrics_scraper`, 3
- `spotichart.lyrics.main`, 4
- `spotichart.lyrics.track_features`, 4
- `spotichart.spotipy.audio_features`, 5
- `spotichart.spotipy.main`, 5
- `spotichart.spotipy.top_charts`, 6

D

`detect_language()` (in *module*
spotichart.language.main), 3

G

`generate_top_chart()` (in *module*
spotichart.spotipy.main), 5

`get_audio_features()` (in *module*
spotichart.spotipy.audio_features), 5

`get_chart()` (in *module*
spotichart.spotipy.top_charts), 6

`get_charts()` (in *module*
spotichart.spotipy.top_charts), 6

`get_lyrics()` (in *module spotichart.lyrics.main*), 4

`get_lyrics_from_chart()` (in *module*
spotichart.lyrics.main), 4

R

`request_song_info()` (in *module*
spotichart.lyrics.track_features), 4

S

`scrap_lyrics()` (in *module*
spotichart.lyrics.lyrics_scraper), 3

`search_song()` (in *module*
spotichart.lyrics.track_features), 5

spotichart (module), 7

spotichart.language (module), 3

spotichart.language.main (module), 3

spotichart.lyrics.lyrics_scraper (mod-
ule), 3

spotichart.lyrics.main (module), 4

spotichart.lyrics.track_features (mod-
ule), 4

spotichart.spotipy.audio_features (mod-
ule), 5

spotichart.spotipy.main (module), 5

spotichart.spotipy.top_charts (module), 6