

# Package ‘DSpoty’

October 12, 2022

**Type** Package

**Date** 2020-01-07

**Title** Get 'Spotify' API Multiple Information

**Version** 0.1.0

**Author** Alberto Almuiña <albertogonzalezalmuinha@gmail.com>

**Maintainer** Alberto Almuiña <albertogonzalezalmuinha@gmail.com>

**Description** You can retrieve 'Spotify' API Information such as artists, albums, tracks, features tracks, recommendations or related artists.

This package allows you to search all the information by name and also includes a distance based algorithm to find similar songs.

More information: <<https://developer.spotify.com/documentation/web-api/>> .

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**URL** <https://github.com/AlbertoAlmuinha/DSpoty>

**Imports** dplyr, purrr, httr, stringr

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2020-01-16 10:50:02 UTC

## R topics documented:

get_album_popularity . . . . .	2
get_artist . . . . .	3
get_artists_by_genre . . . . .	3
get_artist_albums . . . . .	4
get_artist_tracks . . . . .	5
get_artist_tracks_features . . . . .	6
get_related_artists . . . . .	7

get_several_artists . . . . .	7
get_several_tracks . . . . .	8
get_several_tracks_features . . . . .	9
get_several_tracks_features_vectorized . . . . .	10
get_similar_songs_algorithm . . . . .	10
get_spotify_access_token . . . . .	11
get_spotify_item . . . . .	12
get_tracks_recommendations . . . . .	13
get_tracks_recommendations_by_name . . . . .	14
get_track_popularity . . . . .	14

## Index 16

---

get\_album\_popularity *Get Album Popularity*

---

### Description

This function returns the popularity of an album on Spotify.

### Usage

```
get_album_popularity(album_uri, limit = 1, offset = 0,
  access_token = DSpoty::get_spotify_access_token())
```

### Arguments

album_uri	An album uri from Spotify to search for or a group of albums uris separated by a comma
limit	Integer indicating the max number of artists to return. Defaults to 1, max of 50.
offset	Integer indicating the offset of the first artist to return. Defaults to 0 (Spotify's API default value).
access_token	Spotify Web API token. Defaults to DSpoty::get_spotify_access_token()

### Value

Returns an integer vector with the popularity value (0-100).

### Author(s)

Alberto Almui?a

### Examples

```
## Not run:
get_album_popularity('1T2uKSyUf7DBWOABtsmYLM')

## End(Not run)
```

---

get_artist	<i>Get Artist</i>
------------	-------------------

---

**Description**

This function searches spotify artists by name

**Usage**

```
get_artist(artist_name, offset = "0", limit = "1",
           access_token = DSpoty::get_spotify_access_token())
```

**Arguments**

artist_name	String of artist name
offset	Integer indicating the offset of the first artist to return. Defaults to 0 (Spotify's API default value).
limit	Integer indicating the max number of artists to return. Defaults to 1, max of 50.
access_token	Spotify Web API token. Defaults to DSpoty::get_spotify_access_token()

**Value**

Returns a data frame with the artist's information

**Author(s)**

Alberto Almui?a

**Examples**

```
## Not run:
get_artist('Arctic Monkeys')

## End(Not run)
```

---

get_artists_by_genre	<i>Get Artists by Genre</i>
----------------------	-----------------------------

---

**Description**

Function to search artists by genre

**Usage**

```
get_artists_by_genre(genre = is.character(), offset = "0",
                    limit = "10", year = 2019,
                    access_token = DSpoty::get_spotify_access_token())
```

**Arguments**

genre	String of genre name to search for
offset	Integer indicating the offset of the first artist to return. Defaults to 0 (Spotify's API default value).
limit	Integer indicating the max number of artists to return. Defaults to 10, max of 50.
year	Year to search for
access_token	Spotify Web API token. Defaults to DSpoty::get_spotify_access_token()

**Value**

Returns a data frame with the artist's information of a determined genre

**Author(s)**

Alberto Almui?a

**Examples**

```
## Not run:
rock_artists <- get_artists_by_genre("rock")

## End(Not run)
```

---

get_artist_albums	<i>Get Artist Albums</i>
-------------------	--------------------------

---

**Description**

This function returns an artist's discography on Spotify.

**Usage**

```
get_artist_albums(artist_name, album_type = "album", limit = 50,
  offset = 0, access_token = DSpoty::get_spotify_access_token())
```

**Arguments**

artist_name	String of artist name.
album_type	Character vector of album types to include. Valid values are "album", "single", "appears_on", and "compilation". Defaults to "album".
limit	Integer indicating the max number of artists to return. Defaults to 50, max of 50.
offset	Integer indicating the offset of the first artist to return. Defaults to 0 (Spotify's API default value).
access_token	Spotify Web API token. Defaults to DSpoty::get_spotify_access_token()

**Value**

Returns a data frame with all the albums of the selected artist

**Author(s)**

Alberto Almuiña

**Examples**

```
## Not run:  
albums <- get_artist_albums('radiohead')  
  
## End(Not run)
```

---

*get\_artist\_tracks*      *Get Artist Tracks*

---

**Description**

This function returns all the tracks of an artist.

**Usage**

```
get_artist_tracks(artist_name, limit = 50, offset = 0,  
  access_token = DSpoty::get_spotify_access_token())
```

**Arguments**

- `artist_name`      String of artist name.
- `limit`            Integer indicating the max number of artists to return. Defaults to 50, max of 50.
- `offset`           Integer indicating the offset of the first artist to return. Defaults to 0 (Spotify's API default value).
- `access_token`    Spotify Web API token. Defaults to `DSpoty::get_spotify_access_token()`.

**Value**

Returns a data frame with all the tracks of the selected artist.

**Author(s)**

Alberto Almuiña

**Examples**

```
## Not run:  
tracks <- get_artist_tracks('Shakira')  
  
## End(Not run)
```

---

`get_artist_tracks_features`*Get Artist Tracks Features*

---

**Description**

This function returns audio features for all the tracks of a given artist

**Usage**

```
get_artist_tracks_features(artist_name, limit = 100, offset = 0,  
  access_token = DSpoty::get_spotify_access_token())
```

**Arguments**

<code>artist_name</code>	String of an artist name
<code>limit</code>	Integer indicating the max number of artists to return. Defaults to 100, max of 100.
<code>offset</code>	Integer indicating the offset of the first artist to return. Defaults to 0 (Spotify's API default value).
<code>access_token</code>	Spotify Web API token. Defaults to <code>DSpoty::get_spotify_access_token()</code>

**Value**

Returns a data frame with the tracks features for all the tracks of an artist

**Author(s)**

Alberto Almui?a

**Examples**

```
## Not run:  
tracks_features<-get_artist_tracks_features('Shakira')  
  
## End(Not run)
```

---

get\_related\_artists     *Get Related Artists*

---

**Description**

This function returns the most similar artists of the selected artist.

**Usage**

```
get_related_artists(artist_uri,  
                    access_token = DSpoty::get_spotify_access_token())
```

**Arguments**

artist\_uri     Artist ID.  
access\_token     Spotify Web API token. Defaults to DSpoty::get\_spotify\_access\_token()

**Value**

Returns a data frame with the related artist information.

**Author(s)**

Alberto Almui?a

**Examples**

```
## Not run:  
get_related_artists('7Ln80lUS6He07XvHI8qqHH')  
  
## End(Not run)
```

---

get\_several\_artists     *Get Several Artists*

---

**Description**

This function allows you to load several artists in a faster way.

**Usage**

```
get_several_artists(artist_ids_df, ids_label = 1,  
                    access_token = DSpoty::get_spotify_access_token())
```

**Arguments**

artist\_ids\_df    Dataframe containing the artists ids in one of the columns.  
ids\_label        Integer indicating the column number of the artists ids. Default to 1.  
access\_token    Spotify Web API token. Defaults to DSpoty::get\_spotify\_access\_token()

**Value**

Returns a data frame with all the artist's information

**Author(s)**

Alberto Almui?a

**Examples**

```
## Not run:  
get_several_artists(df,1)  
  
## End(Not run)
```

---

get\_several\_tracks    *Get Several Tracks*

---

**Description**

This function allows you to load several tracks in a faster way.

**Usage**

```
get_several_tracks(tracks_id_df, ids_label = 1,  
                  access_token = DSpoty::get_spotify_access_token())
```

**Arguments**

tracks\_id\_df    Dataframe containing the tracks ids in one of the columns.  
ids\_label        Integer indicating the column number of the tracks ids. Default to 1.  
access\_token    Spotify Web API token. Defaults to DSpoty::get\_spotify\_access\_token()

**Value**

Returns a data frame with all the artist's information

**Author(s)**

Alberto Almui?a



### Examples

```
## Not run:  
get_several_tracks(df,1)  
  
## End(Not run)
```

---

`get_several_tracks_features`  
*Get Several Tracks Features*

---

### Description

This function allows you to load several tracks features in a faster way.

### Usage

```
get_several_tracks_features(tracks_df,  
  access_token = DSpoty::get_spotify_access_token())
```

### Arguments

`tracks_df`      Dataframe containing the tracks ids in the first column.  
`access_token`   Spotify Web API token. Defaults to `DSpoty::get_spotify_access_token()`.

### Value

Returns a data frame with all the `tracks_features` for the selected tracks uris.

### Author(s)

Alberto Almui?a

### Examples

```
## Not run:  
get_several_tracks_features(df)  
  
## End(Not run)
```

---

`get_several_tracks_features_vectorized`  
*Get Several Tracks Features Vectorized*

---

**Description**

This function allows you to load several tracks features really fast specially with a large amount of tracks. Improves the performance in comparison with `get_several_tracks_features`.

**Usage**

```
get_several_tracks_features_vectorized(tracks_df,  
    access_token = DSpoty::get_spotify_access_token())
```

**Arguments**

`tracks_df`        Dataframe containing the tracks ids in the first column.  
`access_token`    Spotify Web API token. Defaults to `DSpoty::get_spotify_access_token()`.

**Value**

Returns a data frame with all the `tracks_features` for the selected tracks uris.

**Author(s)**

Alberto Almui?a

**Examples**

```
## Not run:  
get_several_tracks_features_vectorized(df)  
  
## End(Not run)
```

---

`get_similar_songs_algorithm`  
*Get Similar Songs Algorithm*

---

**Description**

This function finds the most similar songs based on related artists.

**Usage**

```
get_similar_songs_algorithm(artist_name, artist_track,  
    access_token = DSpoty::get_spotify_access_token())
```

**Arguments**

artist\_name     String of artist name  
artist\_track    String of artst track  
access\_token    Spotify Web API token. Defaults to DSpoty::get\_spotify\_access\_token()

**Value**

Returns a data frame with the tracks information

**Author(s)**

Alberto Almuiña

**Examples**

```
## Not run:  
get_similar_songs_algorithm('Leiva', 'Nuclear')  
  
## End(Not run)
```

---

*get\_spotify\_access\_token*  
*Get Spotify Access Token.*

---

**Description**

This function returns the Spotify's access token.

**Usage**

```
get_spotify_access_token(client_id = Sys.getenv("client_id"),  
                          client_secret = Sys.getenv("client_secret"))
```

**Arguments**

client\_id       String with the Spotify CLIENT\_ID.  
client\_secret   String with the Spotify CLIENT\_SECRET.

**Value**

Returns a string with de access token.

**Author(s)**

Alberto Almuiña

## Examples

```
## Not run:
access_token<-get_spotify_access_token(client_id = 'client_id', client_secret = 'client_secret')

## End(Not run)
```

---

get_spotify_item	<i>Get Spotify Item</i>
------------------	-------------------------

---

## Description

This function searches spotify item by name

## Usage

```
get_spotify_item(name = NULL, type, limit = 1, offset = 0,
  access_token = DSpoty::get_spotify_access_token(), artist = NULL,
  album = NULL)
```

## Arguments

name	String of item name (artist, album, track...). Default: NULL
type	String of the search type. Possible values are 'artist', 'album' and 'track'.
limit	Integer indicating the max number of artists to return. Defaults to 1, max of 50.
offset	Integer indicating the offset of the first artist to return. Defaults to 0 (Spotify's API default value).
access_token	Spotify Web API token. Defaults to DSpoty::get_spotify_access_token()
artist	String of album's artist name. Default: NULL
album	String of album name. Default: NULL

## Details

You have two options to use this function: Search by parameter name or search by parameters artist + album. In this second case, type must always be 'album'.

## Value

Returns a data frame with the item's information

## Author(s)

Alberto Almuiña

## Examples

```
## Not run:
get_spotify_item(name = 'Arctic Monkeys', type = 'artist')
get_spotify_item(artist = 'Leiva', album = 'Nuclear', type = 'album')
get_spotify_item(name = 'Do I Wanna Know?', type = 'track')

## End(Not run)
```

---

get\_tracks\_recommendations  
*Get Tracks Recommendations*

---

## Description

This function returns the recommended songs by Spotify.

## Usage

```
get_tracks_recommendations(artist_uri, track_uri, limit = 10,
  access_token = DSpoty::get_spotify_access_token())
```

## Arguments

artist_uri	Artist ID.
track_uri	Track ID.
limit	Integer indicating the max number of artists to return. Default 10.
access_token	Spotify Web API token. Defaults to DSpoty::get_spotify_access_token().

## Value

Returns a data frame with the recommended artists information.

## Author(s)

Alberto Almui?a

## Examples

```
## Not run:
get_tracks_recommendations('7Ln80lUS6He07XvHI8qqHH', '0FgNSsaSZTvbLXUumS08LQ')

## End(Not run)
```

---

```
get_tracks_recommendations_by_name
```

*Get Tracks Recommendations By Name*

---

**Description**

This function returns the recommended songs by Spotify.

**Usage**

```
get_tracks_recommendations_by_name(artist_name, track_name, limit = 10,
  access_token = DSpoty::get_spotify_access_token())
```

**Arguments**

artist_name	Artist Name character.
track_name	Track Name character.
limit	Integer indicating the max number of artists to return. Default 10.
access_token	Spotify Web API token. Defaults to DSpoty::get_spotify_access_token().

**Value**

Returns a data frame with the recommended songs information.

**Author(s)**

Alberto Almui?a

**Examples**

```
## Not run:
get_tracks_recommendations_by_name('Arctic Monkeys', 'Do I wanna know?')

## End(Not run)
```

---

```
get_track_popularity
```

*Get Track Popularity*

---

**Description**

This function that returns the popularity of a track uri on Spotify.

**Usage**

```
get_track_popularity(track_uri, limit = 1, offset = 0,
  access_token = DSpoty::get_spotify_access_token())
```

**Arguments**

<code>track_uri</code>	A track uri corresponding to one track in Spotify.
<code>limit</code>	Integer indicating the max number of artists to return. Defaults to 1, max of 50.
<code>offset</code>	Integer indicating the offset of the first artist to return. Defaults to 0 (Spotify's API default value).
<code>access_token</code>	Spotify Web API token. Defaults to <code>DSPoty::get_spotify_access_token()</code> .

**Value**

Returns an integer vector with the popularity value (0-100).

**Author(s)**

Alberto Almui?a

**Examples**

```
## Not run:  
popularity <- get_track_popularity('0FgNSsaSZTvbLXUumS08LQ')  
  
## End(Not run)
```

# Index

[get\\_album\\_popularity](#), 2  
[get\\_artist](#), 3  
[get\\_artist\\_albums](#), 4  
[get\\_artist\\_tracks](#), 5  
[get\\_artist\\_tracks\\_features](#), 6  
[get\\_artists\\_by\\_genre](#), 3  
[get\\_related\\_artists](#), 7  
[get\\_several\\_artists](#), 7  
[get\\_several\\_tracks](#), 8  
[get\\_several\\_tracks\\_features](#), 9  
[get\\_several\\_tracks\\_features\\_vectorized](#),  
10  
[get\\_similar\\_songs\\_algorithm](#), 10  
[get\\_spotify\\_access\\_token](#), 11  
[get\\_spotify\\_item](#), 12  
[get\\_track\\_popularity](#), 14  
[get\\_tracks\\_recommendations](#), 13  
[get\\_tracks\\_recommendations\\_by\\_name](#), 14