

# Package ‘geniusr’

April 3, 2019

**Title** Tools for Working with the 'Genius' API

**Version** 1.1.0

**Description** Provides tools to interact nicely with the 'Genius' API (<https://docs.genius.com/>). Search hosted content, extract different types of associated metadata and retrieve lyrics with ease.

**URL** <https://github.com/ewenme/geniusr>

**BugReports** <https://github.com/ewenme/geniusr/issues>

**Depends** R ( $\geq$  3.2.0)

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** yes

**RoxygenNote** 6.1.1

**Imports** httr, purrr, tibble, rvest, stringr, xml2, curl, attempt

**NeedsCompilation** no

**Author** Ewen Henderson [aut, cre]

**Maintainer** Ewen Henderson <[ewenhenderson@gmail.com](mailto:ewenhenderson@gmail.com)>

**Repository** CRAN

**Date/Publication** 2019-01-20 21:50:02 UTC

## R topics documented:

geniusr . . . . .	1
genius.token . . . . .	2
get_album_meta . . . . .	2
get_artist_meta . . . . .	3
get_artist_songs . . . . .	3
get_song_meta . . . . .	4
scrape_lyrics_id . . . . .	4
scrape_lyrics_url . . . . .	5
scrape_tracklist . . . . .	5
search_artist . . . . .	6
search_song . . . . .	6

---

geniusr	geniusr package
---------	-----------------

---

### Description

An interface to Genius' Web API.

### Details

See the README on GitHub<sup>1</sup>

---

genius_token	Get or set <i>GENIUS_API_TOKEN</i> value
--------------	--

---

### Description

The API wrapper functions in this package all rely on a Genius client access token residing in the environment variable `GENIUS_API_TOKEN`. The easiest way to accomplish this is to set it in the `‘.Renviron’` file in your home directory.

### Usage

```
genius_token(force = FALSE)
```

### Arguments

force	force setting a new Genius API token for the current environment?
-------	---

### Value

atomic character vector containing the Genius API token

---

get_album_meta	Retrieve meta data for an album
----------------	---------------------------------

---

### Description

The Genius API lets you search for an album's meta data, given an album ID.

### Usage

```
get_album_meta(album_id, access_token = genius_token())
```

### Arguments

album_id	An album ID (album_id returned in <code>get_song_meta</code> )
access_token	Genius' client access token, defaults to <code>genius_token</code>

---

<sup>1</sup><https://github.com/ewenme/geniusr#readme>

### Examples

```
## Not run:
get_album_meta(album_id = 337082)

## End(Not run)
```

---

get_artist_meta	<i>Retrieve meta data for an artist</i>
-----------------	---

---

### Description

The Genius API lets you search for meta data for an artist, given an artist ID.

### Usage

```
get_artist_meta(artist_id, access_token = genius_token())
```

### Arguments

**artist\_id**      An artist ID (**artist\_id** returned in **search\_artist**)  
**access\_token**   Genius' client access token, defaults to **genius\_token**

### Examples

```
## Not run:
get_artist_meta(artist_id = 16751)

## End(Not run)
```

---

get_artist_songs	<i>Retrieve meta data for all of an artist's appearances on Genius</i>
------------------	--

---

### Description

Return meta data for all appearances (features optional) of an artist on Genius.

### Usage

```
get_artist_songs(artist_id, include_features = FALSE,
  access_token = genius_token())
```

### Arguments

**artist\_id**      An artist ID (**artist\_id** returned in **search\_artist**)  
**include\_features**      Whether to return results where artist isn't the primary artist (logical, defaults to FALSE)  
**access\_token**   Genius' client access token, defaults to **genius\_token**

## Examples

```
## Not run:
get_artist_songs(artist_id = 1421)

## End(Not run)
```

---

get_song_meta	<i>Retrieve meta data for a song</i>
---------------	--------------------------------------

---

## Description

The Genius API lets you search for meta data for a song, given a song ID.

## Usage

```
get_song_meta(song_id, access_token = genius_token())
```

## Arguments

song_id	A song ID (song_id returned in <code>search_song</code> )
access_token	Genius' client access token, defaults to <code>genius_token</code>

## Examples

```
## Not run:
get_song_meta(song_id = 3039923)

## End(Not run)
```

---

scrape_lyrics_id	<i>Retrieve lyrics associated with a Genius song ID</i>
------------------	---

---

## Description

Scrape lyrics from Genius' lyric pages using an associated song ID.

## Usage

```
scrape_lyrics_id(song_id, access_token = genius_token())
```

## Arguments

song_id	song ID (like in <code>song_id</code> returned by <code>search_song</code> )
access_token	Genius' client access token, defaults to <code>genius_token</code>

## Examples

```
## Not run:
scrape_lyrics_id(song_id = 3214267)

## End(Not run)
```

---

scrape_lyrics_url	<i>Retrieve lyrics associated with a Genius lyrics page URL</i>
-------------------	---

---

### Description

Scrape lyrics from a Genius' lyric page using it's associated URL. Best used with `scrape_tracklist`, when song IDs aren't returned - otherwise, `scrape_lyrics_id` is recommended.

### Usage

```
scrape_lyrics_url(song_lyrics_url, access_token = genius_token())
```

### Arguments

song_lyrics_url	song lyrics url (like in <code>song_lyrics_url</code> returned by <code>get_song_meta</code> )
access_token	Genius' client access token, defaults to <code>genius_token</code>

### Examples

```
## Not run:  
scrape_lyrics_url(song_lyrics_url = "https://genius.com/Kendrick-lamar-dna-lyrics")  
  
## End(Not run)
```

---

scrape_tracklist	<i>Retrieve an album's tracklisting</i>
------------------	---

---

### Description

Scrape an album's tracklisting, and song meta data, given an album ID.

### Usage

```
scrape_tracklist(album_id, access_token = genius_token())
```

### Arguments

album_id	An album ID ( <code>album_id</code> returned in <code>get_song_meta</code> )
access_token	Genius' client access token, defaults to <code>genius_token</code>

### Examples

```
## Not run:  
scrape_tracklist(album_id = 337082)  
  
## End(Not run)
```

---

<code>search_artist</code>	<i>Retrieve artist identifiers for associated search terms</i>
----------------------------	--

---

### Description

The Genius API lets you search hosted content (all songs). Use `search_artist` to return `artist_id`, `artist_name` and `artist_url` for all unique artist matches to a search.

### Usage

```
search_artist(search_term, n_results = 10,
              access_token = genius_token())
```

### Arguments

<code>search_term</code>	A character string to search for artist matches
<code>n_results</code>	Maximum no. of search results to return (this is the number of hosted content search results, unique artist matches will be smaller)
<code>access_token</code>	Genius' client access token, defaults to <code>genius_token</code>

### Examples

```
## Not run:
search_artist(search_term = "Lil")

## End(Not run)
```

---

<code>search_song</code>	<i>Retrieve song identifiers for associated search terms</i>
--------------------------	--

---

### Description

The Genius API lets you search hosted content (all songs). Use `search_song` to return `song_id`, `song_name`, `lyrics_url` and `artist_id` for all unique song matches to a search.

### Usage

```
search_song(search_term, n_results = 10, access_token = genius_token())
```

### Arguments

<code>search_term</code>	A character string to search for song matches
<code>n_results</code>	Maximum no. of search results to return
<code>access_token</code>	Genius' client access token, defaults to <code>genius_token</code>

### Examples

```
## Not run:
search_song(search_term = "Gucci", n_results=50)

## End(Not run)
```