Package ‘geniusr’

October 13, 2022

Title Tools for Working with the 'Genius' API

Version 1.2.1

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

https://github.com/ewenme/geniusr

BugReports https://github.com/ewenme/geniusr/issues

Depends R (>= 3.2.0)
License MIT + file LICENSE
Imports curl, dplyr, httr, purrr, rvest, stringr, tibble, xml2
Suggests knitr, rmarkdown, testthat (>= 2.1.0), covr

Encoding UTF-8
RoxygenNote 7.2.1

NeedsCompilation no

Author Ewen Henderson [aut, cre] (<https://orcid.org/0000-0002-4748-4693>)
Maintainer Ewen Henderson <ewenhenderson@gmail.com>

Repository CRAN

Date/Publication 2022-09-03 22:10:02 UTC

R topics documented:

album_to_df .............................................................. 2
artist_to_df .............................................................. 3
browse_genius .............................................................. 3
browse_genius_resource ..................................................... 4
genius_token .............................................................. 5
get_album ................................................................. 5
get_album_df ............................................................... 6
get_album_tracklist_id ...................................................... 7
Index

album_to_df Convert genius_album object to a data frame

Description
Convert genius_album object to a data frame

Usage
album_to_df(x)

Arguments
x a genius_album object

Value
a tibble
**artist_to_df**

**Examples**

```r
## Not run:
album <- get_album(album_id = 337082)
album_to_df(album)

## End(Not run)
```

**Description**

Convert genius_artist object to a data frame

**Usage**

```r
artist_to_df(x)
```

**Arguments**

- `x`: a genius_artist object

**Value**

a tibble

**Examples**

```r
## Not run:
artist <- get_artist(artist_id = 16775)
artist_to_df(artist)

## End(Not run)
```

---

**browse_genius**

Open the Genius homepage in your browser

**Description**

Opens a browser to https://genius.com/.

**Usage**

```r
browse_genius()
```
Value

A browser is opened to the Genius website if the session is interactive. The URL is returned as a character string.

Examples

```r
browse_genius()
```

---

browse_genius_resource

Open the Genius url of a resource in your browser

Description

Opens a browser to the Genius url of a Genius "resource" (i.e. the result of a successful `genius_album` | `genius_artist` | `genius_song` call).

Usage

```r
browse_genius_resource(x)
```

Arguments

- `x` : a `genius_album`, `genius_artist`, or `genius_song` object

Value

A browser is opened to the Genius resource’s url if the session is interactive. The URL is returned as a character string.

Examples

```r
song <- get_song(song_id = 3039923)
browse_genius_resource(song)
```
### genius_token

**Get or set Genius access token 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**

```r
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

**Retrieve metadata for an album**

**Description**

The Genius API lets you request data for a specific album, given an album ID. `get_album()` returns this data in full.

**Usage**

```r
get_album(album_id, access_token = genius_token())
```

**Arguments**

- **album_id**
  
  ID of the album (`album_id` within an object returned by `get_song`)

- **access_token**
  
  Genius’ client access token, defaults to `genius_token`

**Value**

a `genius_album` object that contains the extracted content from the request, the original JSON response object and the request path.

**See Also**

See `get_album_df` to return a tidy data frame.

Other album: `get_album_df`, `get_album_tracklist_id`, `get_album_tracklist_search()`
get_album_df

Retrieve meta data for an album

Description

The Genius API lets you return data for a specific album, given an album ID. get_album_meta returns this data in a tidy, but reduced, format.

Usage

get_album_df(album_id, access_token = genius_token())

Arguments

album_id ID of the album (album_id within an object returned by get_song)
access_token Genius’ client access token, defaults to genius_token

Value

a tibble

See Also

See get_album to return extended data as a list.

Other album: get_album_tracklist_id(), get_album_tracklist_search(), get_album()

Examples

## Not run:
get_album(album_id = 337082)

## End(Not run)
get_album_tracklist_id

Retrieve an album's tracklisting

Description
Get an album’s tracklisting, and song meta data, given an album ID.

Usage
get_album_tracklist_id(album_id, access_token = genius_token())

Arguments
- album_id: ID of the album (album_id within an object returned by get_song)
- access_token: Genius’ client access token, defaults to genius_token

Value
a tibble

See Also
See get_album_tracklist_search to search for an album tracklist by searching artist/album names.

Other album: get_album_df(), get_album_tracklist_search(), get_album()

Examples
## Not run:
get_album_tracklist_id(album_id = 337082)
## End(Not run)

get_album_tracklist_search

Retrieve an album’s tracklisting

Description
Attempt to get an album’s tracklisting, given an artist and album name.

Usage
get_album_tracklist_search(artist_name, album_name)
get_annotation

Arguments

- **artist_name**: Name of artist
- **album_name**: Name of album

Value

- a tibble

See Also

See **get_album_tracklist_id** to search for an album tracklist using an album ID.

Other album: **get_album_df()**, **get_album_tracklist_id()**, **get_album()**

Examples

```r
## Not run:
get_album_tracklist_search(artist_name = "Kendrick Lamar",
album_name = "DAMN.")
## End(Not run)
```

get_annotation

Retrieve metadata for an annotation

Description

The Genius API lets you return data for a specific annotation, given an annotation ID. get_annotation returns this data in full.

Usage

```r
get_annotation(annotation_id, access_token = genius_token())
```

Arguments

- **annotation_id**: ID of the annotation
- **access_token**: Genius’ client access token, defaults to `genius_token`

Details

A Genius annotation is a piece of content about a part of a document. The document may be a song (hosted on Genius) or a web page (hosted anywhere). The part of a document that an annotation is attached to is called a referent.

Value

- a `genius_annotation` object that contains the extracted content from the request, the original JSON response object and the request path.
get_artist

See Also

Other annotation: get_referent()

Examples

```r
## Not run:
get_annotation(annotation_id = 16511101)
## End(Not run)
```

---

### Description

The Genius API lets you return data for a specific artist, given an artist ID. `get_artist` returns this data in full.

### Usage

```r
get_artist(artist_id, access_token = genius_token())
```

### Arguments

- **artist_id**: ID of the artist (`artist_id` within an object returned by `search_artist`)
- **access_token**: Genius’ client access token, defaults to `genius_token`

### Value

A `genius_artist` object that contains the extracted content from the request, the original JSON response object and the request path.

### See Also

See `get_artist_df` to return a tidy data frame.

Other artist: `get_artist_df()`, `get_artist_songs_df()`, `get_artist_songs()`

### Examples

```r
## Not run:
get_artist(artist_id = 16775)
## End(Not run)
```
**get_artist_df**  
*Retrieve metadata for an artist*

Description

The Genius API lets you search for meta data for an artist, given an artist ID. `get_artist_df` returns this data in a tidy, but reduced, format.

Usage

```r
get_artist_df(artist_id, access_token = genius_token())
```

Arguments

- `artist_id`  
  ID of the artist (artist_id within an object returned by `search_artist`)
- `access_token`  
  Genius’ client access token, defaults to `genius_token`

Value

a tibble

See Also

See `get_artist` to return data in full as a list.

Other artist: `get_artist_songs_df`, `get_artist_songs`, `get_artist`

Examples

```r
## Not run:
get_artist_df(artist_id = 16751)
## End(Not run)
```

**get_artist_songs**  
*Retrieve metadata for all of an artist’s songs*

Description

The Genius API lets you search for song metadata of an artist, given an artist ID. `get_artist_songs` returns this data in full.
get_artist_songs_df

Usage

get_artist_songs(
  artist_id,
  sort = c("title", "popularity"),
  include_features = FALSE,
  access_token = genius_token()
)

Arguments

artist_id      ID of the artist (artist_id within an object returned by search_artist)
sort          method to order results; by "title" (default) or by "popularity"
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

Value

a genius_resource object that contains the extracted content from the request, the original JSON response object and the request path.

See Also

See get_artist_songs_df to return a tidy data frame.

Other artist: get_artist_df(), get_artist_songs_df(), get_artist()

Examples

## Not run:
get_artist_songs(artist_id = 1421)
get_artist_songs(artist_id = 1421, sort = "popularity")
get_artist_songs(artist_id = 1421, include_features = TRUE)

## End(Not run)

get_artist_songs_df    Retrieve metadata for all of an artist's songs

Description

The Genius API lets you search for song metadata of an artist, given an artist ID. get_artist_songs_df returns this data in a tidy, but reduced, format.
get_lyrics_id

Usage

get_artist_songs_df(
  artist_id,
  sort = c("title", "popularity"),
  include_features = FALSE,
  access_token = genius_token()
)

Arguments

artist_id ID of the artist (artist_id within an object returned by search_artist)
sort method to order results; by "title" (default) or by "popularity"
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

Value

a tibble

See Also

See get_artist_songs to return data in full as a list.
Other artist: get_artist_df(), get_artist_songs(), get_artist()

Examples

## Not run:
get_artist_songs_df(artist_id = 1421)

## End(Not run)

get_lyrics_id Retrieve lyrics associated with a Genius song ID

Description

Get lyrics from Genius’ lyric pages using an associated song ID.

Usage

get_lyrics_id(song_id, access_token = genius_token())
get_lyrics_search

Arguments

- song_id: ID of the song (song_id within an object returned by \texttt{search_song})
- access_token: Genius’ client access token, defaults to \texttt{genius_token}

See Also

See \texttt{get_lyrics_url} to search lyrics using a song URL, \texttt{get_lyrics_search} to search using artist name and song title.

Other lyrics: \texttt{get_lyrics_search()}, \texttt{get_lyrics_url()}

Examples

```r
## Not run:
get_lyrics_id(song_id = 3214267)
## End(Not run)
```

---

\begin{itemize}
  \item \textit{get_lyrics_search} \hspace{1cm} Retrieve lyrics associated with a Genius song
\end{itemize}

Description

Attempt to get lyrics from Genius’ lyric pages using an associated artist name and song title.

Usage

\begin{verbatim}
get_lyrics_search(artist_name, song_title)
\end{verbatim}

Arguments

- artist_name: Name of artist
- song_title: Title of song

See Also

See \texttt{get_lyrics_id} to search lyrics using a song ID, \texttt{get_lyrics_url} to search using a song URL.

Other lyrics: \texttt{get_lyrics_id()}, \texttt{get_lyrics_url()}

Examples

```r
## Not run:
get_lyrics_search(artist_name = "Anderson .Paak", song_title = "Come Home")
## End(Not run)
```
get_lyrics_url

Retrieve lyrics associated with a Genius lyrics page URL

Description
Scrape lyrics from a Genius’ lyric page using it’s associated URL. Best used with get_album_tracklist_id, when song IDs aren’t returned - otherwise, get_lyrics_id is recommended.

Usage
get_lyrics_url(song_lyrics_url)

Arguments
song_lyrics_url
song lyrics url (like in song_lyrics_url returned by get_song_df)

See Also
See get_lyrics_id to search lyrics using a song ID, get_lyrics_search to search using artist name and song title
Other lyrics: get_lyrics_id(), get_lyrics_search()

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

get_referent

Retrieve metadata for a referent

Description
The Genius API lets you return data for a specific referent. get_referent returns this data in full.

Usage
get_referent(
    created_by_id,
    song_id,
    web_page_id,
    access_token = genius_token()
)
**get_song**

**Arguments**

- **created_by_id**: ID of a user to get referents for
- **song_id**: ID of a song to get referents for (pass only one of `song_id` and `web_page_id`)
- **web_page_id**: ID of a web page to get referents for (pass only one of `song_id` and `web_page_id`)
- **access_token**: Genius’ client access token, defaults to `genius_token`

**Details**

Referents are the sections of a piece of content to which annotations are attached. Each referent is associated with a web page or a song and may have one or more annotations. Referents can be searched by the document they are attached to or by the user that created them.

**Value**

A `genius_referent` object that contains the extracted content from the request, the original JSON response object and the request path.

**See Also**

Other annotation: `get_annotation()`

**Examples**

```r
## Not run:
get_referent(song_id = 3039923)
## End(Not run)
```

---

**get_song**  
*Retrieve metadata for a song*

**Description**

The Genius API lets you return data for a specific song, given a song ID. `get_song` returns this data in full.

**Usage**

```r
get_song(song_id, access_token = genius_token())
```

**Arguments**

- **song_id**: ID of the song (`song_id` within an object returned by `search_song`)
- **access_token**: Genius’ client access token, defaults to `genius_token`
Value

a **genius_song** object that contains the extracted content from the request, the original JSON response object and the request path.

See Also

See **get_song_df** to return a tidy data frame.

Other song: **get_song_df()**, **tidy_album_performances()**, **tidy_song_performances()**, **tidy_song_producers()**, **tidy_song_relationships()**, **tidy_song_writers()**

Examples

```r
## Not run:
get_song(song_id = 3039923)

## End(Not run)
```

---

**get_song_df**

*Retrieve metadata for a song*

Description

The Genius API lets you search for meta data for a song, given a song ID. **get_song_meta** returns this data in a tidy, but reduced, format.

Usage

```r
get_song_df(song_id, access_token = genius_token())
```

Arguments

- **song_id**: ID of the song (song_id within an object returned by **search_song**)
- **access_token**: Genius' client access token, defaults to **genius_token**

Value

a tibble

See Also

See **get_song** to return data in full as a list.

Other song: **get_song()**, **tidy_album_performances()**, **tidy_song_performances()**, **tidy_song_producers()**, **tidy_song_relationships()**, **tidy_song_writers()**
print.genius_album

Examples

## Not run:
get_song_df(song_id = 3039923)

## End(Not run)

print.genius_album  Slightly more human-readable output for genius_album objects

Description

Slightly more human-readable output for genius_album objects

Usage

## S3 method for class 'genius_album'
print(x, ...)

Arguments

x  a genius_album object
...
... ignored

print.genius_annotation  Slightly more human-readable output for genius_annotation objects

Description

Slightly more human-readable output for genius_annotation objects

Usage

## S3 method for class 'genius_annotation'
print(x, ...)

Arguments

x  a genius_annotation object
...
... ignored
**print.genius_artist**  
*Slightly more human-readable output for genius_artist objects*

**Description**

Slightly more human-readable output for genius_artist objects

**Usage**

```r
## S3 method for class 'genius_artist'
print(x, ...)
```

**Arguments**

- `x`: a genius_artist object
- `...`: ignored

---

**print.genius_referent**  
*Slightly more human-readable output for genius_referent objects*

**Description**

Slightly more human-readable output for genius_referent objects

**Usage**

```r
## S3 method for class 'genius_referent'
print(x, ...)
```

**Arguments**

- `x`: a genius_referent object
- `...`: ignored
print.genius_resource  Slightly more human-readable output for genius_resource objects

Description

Slightly more human-readable output for genius_resource objects

Usage

## S3 method for class 'genius_resource'
print(x, ...)

Arguments

x  a genius_resource object
...
ignored

print.genius_song  Slightly more human-readable output for genius_song objects

Description

Slightly more human-readable output for genius_song objects

Usage

## S3 method for class 'genius_song'
print(x, ...)

Arguments

x  a genius_song object
...
ignored
search_artist  Search artists on Genius

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 found using a search term.

Usage

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

Arguments

- search_term: A character string to search for
- n_results: Maximum no. of search results to return
- access_token: Genius' client access token, defaults to genius_token()

Value

a tibble

See Also

Other search: search_genius(), search_song()

Examples

```r
## Not run:
ssearch_artist(search_term = "Lil", n_results = 20)
## End(Not run)
```

search_genius  Search documents hosted on Genius

Description

The Genius API lets you search hosted content (all songs). Use search_genius() to return hits on for a given search term, in full.

Usage

search_genius(search_term, n_results = 10, access_token = genius_token())
**search_song**

**Arguments**

- **search_term** A character string to search for
- **n_results** Maximum no. of search results to return
- **access_token** Genius' client access token, defaults to `genius_token`

**Value**

A `genius_resource` object that contains the extracted content from the request, the original JSON response object and the request path.

**See Also**

Other search: `search_artist()`, `search_song()`

**Examples**

```r
## Not run:
search_genius(search_term = "Lil", n_results = 100)
## End(Not run)
```

---

**search_song** *Search songs on Genius*

**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 found using a search term.

**Usage**

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

**Arguments**

- **search_term** A character string to search for
- **n_results** Maximum no. of search results to return
- **access_token** Genius’ client access token, defaults to `genius_token`

**Value**

A tibble

**See Also**

Other search: `search_artist()`, `search_genius()`
Examples

```r
## Not run:
song <- get_song(song_id = 3039923)
song_to_df(song)
## End(Not run)
```

**song_to_df**

*Convert genius_song object to a data frame*

**Description**

Convert genius_song object to a data frame

**Usage**

```r
song_to_df(x)
```

**Arguments**

- `x` a genius_song object

**Value**

a tibble

**Examples**

```r
## Not run:
song <- get_song(song_id = 3039923)
song_to_df(song)
## End(Not run)
```

**tidy_album_performances**

*Extract album performances from a Genius album*

**Description**

Extract "album performances" (i.e. album credits) info from a Genius album object, as a tidy tibble.

**Usage**

```r
tidy_album_performances(x)
```
tidy_song_performances

Arguments

x A genius_album object

Value

a tibble

See Also

See `get_album` to generate a Genius album object.

Other song: `get_song_df()`, `get_song()`, `tidy_song_performances()`, `tidy_song_producers()`, `tidy_song_relationships()`, `tidy_song_writers()`

Examples

```r
## Not run:
album <- get_album(album_id = 337082)
tidy_album_performances(album)
## End(Not run)
```

### tidy_song_performances

*Extract custom performances from a Genius song*

Description

Extract "custom performances" (i.e. other song credits) info from a Genius song object, as a tidy tibble.

Usage

```
tidy_song_performances(x)
```

Arguments

x A genius_song object

Value

a tibble

See Also

See `get_song` to generate a Genius song object.

Other song: `get_song_df()`, `get_song()`, `tidy_album_performances()`, `tidy_song_producers()`, `tidy_song_relationships()`, `tidy_song_writers()`
tidy_song_producers

Examples

```r
## Not run:
song <- get_song(song_id = 3039923)
tidy_song_performances(song)
## End(Not run)
```

---

**tidy_song_producers**  
*Extract producer credits from a Genius song*

**Description**

Extract "producer artists" (i.e. producer credits) info from a Genius song object, as a tidy tibble.

**Usage**

```r
tidy_song_producers(x)
```

**Arguments**

- `x`  
  A `genius_song` object

**Value**

- A tibble

**See Also**

See `get_song` to generate a Genius song object.

Other song: `get_song_df()`, `get_song()`, `tidy_album_performances()`, `tidy_song_performances()`, `tidy_song_relationships()`, `tidy_song_writers()`

**Examples**

```r
## Not run:
song <- get_song(song_id = 3039923)
tidy_song_producers(song)
## End(Not run)
```
**tidy_song_relationships**

*Extract song relationships from a Genius song*

**Description**

Extract "song relationships" info from a Genius song object, as a tidy tibble.

**Usage**

```r
 tidy_song_relationships(x)
```

**Arguments**

- `x` A `genius_song` object

**Value**

A tibble

**See Also**

See `get_song` to generate a Genius song object.

Other song: `get_song_df()`, `get_song()`, `tidy_album_performances()`, `tidy_song_performances()`, `tidy_song_producers()`, `tidy_song_writers()`

**Examples**

```r
## Not run:
song <- get_song(song_id = 3039923)

tidy_song_relationships(song)

## End(Not run)
```

---

**tidy_song_writers**

*Extract writer credits from a Genius song*

**Description**

Extract "writer artists" (i.e. writer credits) info from a Genius song object, as a tidy tibble.

**Usage**

```r
 tidy_song_writers(x)
```

**Examples**

```r
## Not run:
song <- get_song(song_id = 3039923)

tidy_song_writers(song)

## End(Not run)
```
tidy_song_writers

Arguments

x               A genius_song object

Value

a tibble

See Also

See get_song to generate a Genius song object.

Other song: get_song_df(), get_song(), tidy_album_performances(), tidy_song_performances(), tidy_song_producers(), tidy_song_relationships()

Examples

## Not run:
song <- get_song(song_id = 3039923)
tidy_song_writers(song)

## End(Not run)
Index

* album
  get_album, 5
  get_album_df, 6
  get_album_tracklist_id, 7
  get_album_tracklist_search, 7
* annotation
  get_annotation, 8
  get_referent, 14
* artist
  get_artist, 9
  get_artist_df, 10
  get_artist_songs, 10
  get_artist_songs_df, 11
* lyrics
  get_lyrics_id, 12
  get_lyrics_search, 13
  get_lyrics_url, 14
* search
  search_artist, 20
  search_genius, 20
  search_song, 13
* song
  get_song, 15
  get_song_df, 16
  tidy_album_performances, 22
  tidy_song_performances, 23
  tidy_song_producers, 24
  tidy_song_relationships, 25
  tidy_song_writers, 25

album_to_df, 2
artist_to_df, 3
browse_genius, 3
browse_genius_resource, 4
genius_token, 5
get_album, 5, 6–8, 23
get_album_df, 5, 6, 7, 8
get_album_tracklist_id, 5, 6, 7, 8, 14
get_album_tracklist_search, 5–7, 7
get_annotation, 8, 15
get_artist, 9, 10–12
get_artist_df, 9, 10, 11, 12
get_artist_songs, 9, 10, 10, 12
get_artist_songs_df, 9–11, 11
get_lyrics_id, 12, 13, 14
get_lyrics_search, 13, 13, 14
get_lyrics_url, 13, 14
get_referent, 9, 14
get_song, 5–7, 15, 16, 23–26
get_song_df, 14, 16, 16, 23–26
print.genius_album, 17
print.genius_annotation, 17
print.genius_artist, 18
print.genius_referent, 18
print.genius_resource, 19
print.genius_song, 19
search_artist, 9–12, 20, 21
search_genius, 20, 20, 21
search_song, 13, 15, 16, 20, 21, 21
song_to_df, 22
tidy_album_performances, 16, 22, 23–26
tidy_song_performances, 16, 23, 23, 24–26
tidy_song_producers, 16, 23, 24, 25, 26
tidy_song_relationships, 16, 23, 24, 25, 26
tidy_song_writers, 16, 23–25, 25