
GitSound Documentation

Release 1.0.0

Nicole Woytarowicz, Kylie Dale, Ben Williams, Michael Guida

December 07, 2015

CONTENTS

1	Indices and tables	3
	Python Module Index	5
	Index	7

A program to version control Spotify playlists.

Contents:

class `gitsound.SpotifyUser` (*username, client_id, client_secret, redirect_uri*)

A class that creates a Spotify user “self”

add_song_to_playlist (*uid, pid, songid*)

Function to add song to playlist

Parameters

- **uid** (*string*) – user ID
- **pid** (*string*) – playlist ID to add song to
- **songid** (*string*) – ID of song to add

Raises `RuntimeError`

commit_changes_to_playlist (*uid, pid*)

Function to commit changes to playlist

Parameters

- **uid** (*string*) – user ID
- **pid** (*string*) – playlist ID

get_playlist_from_id (*pid*)

Function that returns playlist name from pid

Parameters **pid** (*string*) – playlist ID

Returns playlist name

get_playlist_id (*position*)

Function to get a single playlist’s ID

Parameters **position** (*string*) – specifies what playlist to get

Returns ID in following format ‘{“pid”: foo, “uid”: bar}’

get_playlist_ids ()

Function to get playlist ids of user

Parameters **self** – Spotify user

Returns list – list of ids in following format ‘{“pid”: foo, “uid”: bar}’

get_playlist_name (*position*)

Function that returns playlist name from position

Parameters **position** (*string*) – specifies what playlist to get name from

Returns playlist name

get_playlist_names ()

Function to get all playlist names

Returns list – of playlist names

get_playlist_tracks (*uid, pid*)

Function to get tracks from pid

Parameters

- **uid** (*string*) – user ID

- **pid** (*string*) – playlist ID

Returns list – tracks on playlist corresponding to pid

init_git_playlist (*uid, pid*)

Function to initialize playlist.

Parameters

- **uid** (*string*) – user ID
- **pid** (*string*) – playlist ID to initialize

Raises RuntimeError

pull_spotify_playlist (*uid, pid*)

Function to pull playlist from Spotify

Parameters

- **uid** (*string*) – user ID
- **pid** (*string*) – playlist ID

Returns string – stating status of pull (either successfull or not)

remove_song_from_playlist (*uid, pid, songid*)

Function to remove song from playlist

Parameters

- **uid** (*string*) – user ID
- **pid** (*string*) – playlist ID to remove song from
- **songid** (*string*) – ID of song to remove

Raises RuntimeError

song_lookup (*name=None, artist=None, limit=1*)

Function to look up song

Parameters

- **name** (*string*) – name of song
- **artist** (*string*) – artist of song
- **limit** (*int*) – max number of results to be returned

Returns dictionary – with track name, artist, and ID

util.check_if_git_playlist (*gitDir, playlistPath*)

Function to check if playlist path is valid

Raises RuntimeError

util.load_config ()

Function to load the config file

Returns the config file

Raises RuntimeError

util.save_config (*data*)

Function to save the config file

Raises RuntimeError

INDICES AND TABLES

- `genindex`

g

gitsound, 1

u

util, 2

A

`add_song_to_playlist()` (`gitsound.SpotifyUser` method), 1

C

`check_if_git_playlist()` (in module `util`), 2

`commit_changes_to_playlist()` (`gitsound.SpotifyUser` method), 1

G

`get_playlist_from_id()` (`gitsound.SpotifyUser` method), 1

`get_playlist_id()` (`gitsound.SpotifyUser` method), 1

`get_playlist_ids()` (`gitsound.SpotifyUser` method), 1

`get_playlist_name()` (`gitsound.SpotifyUser` method), 1

`get_playlist_names()` (`gitsound.SpotifyUser` method), 1

`get_playlist_tracks()` (`gitsound.SpotifyUser` method), 1

`gitsound` (module), 1

I

`init_git_playlist()` (`gitsound.SpotifyUser` method), 2

L

`load_config()` (in module `util`), 2

P

`pull_spotify_playlist()` (`gitsound.SpotifyUser` method), 2

R

`remove_song_from_playlist()` (`gitsound.SpotifyUser` method), 2

S

`save_config()` (in module `util`), 2

`song_lookup()` (`gitsound.SpotifyUser` method), 2

`SpotifyUser` (class in `gitsound`), 1

U

`util` (module), 2