
slacksound Documentation

Release 0.1.1

Oriol Fabregas

Oct 27, 2017

Contents

1	SlackSound	3
1.1	Features	3
1.2	How does it work	3
2	Installation	5
3	Usage	7
3.1	Slack	7
3.2	Spotify	7
3.3	Configuration	7
4	Contributing	9
4.1	Submit Feedback	9
5	slacksound	11
5.1	slacksound package	11
6	Credits	19
6.1	Development Lead	19
6.2	Contributors	19
7	History	21
8	0.0.1 (2017-10-13)	23
9	0.0.2 (2017-10-14)	25
10	0.0.3 (2017-10-14)	27
11	0.0.4 (2017-10-26)	29
12	0.0.5 (2017-10-27)	31
13	0.1.0 (2017-10-27)	33
14	0.1.1 (2017-10-27)	35
15	Indices and tables	37

Contents:

CHAPTER 1

SlackSound

Create playlists democratically by reactions in Slack

!! THIS IS STILL A BETA !!

- Documentation: <https://slacksound.readthedocs.org/en/latest>

Features

- Get Youtube videos reactions from message attachments and add the song to a Spotify playlist.

How does it work

The application acts as a middle man between Slack and Spotify by getting the reactions count from a message's attachment and then looking up the title on Spotify's. If the title has been found, it will add the one that has best popularity to the queue.

In order to the bot pick up the song, a user must paste a Youtube URL in the channel. The bot will start looking for reactions on that URL and get the title of the video which it will be used as a search text for Spotify.

A message will be shown back to the channel whether the bot could add the song to the playlist or not.

It is worth mention that the tracks of the playlist will be removed at every time the bot it is started. This is by design as it in a jukebox.

CHAPTER 2

Installation

At the command line:

```
$ pip install slacksound
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv slacksound  
$ pip install slacksound
```


Slack

The first step is to create a Slack bot. Instructions can be found [here](#) under the title How do I create custom bot users for my workspace?

You can give it a name and assign an avatar to it. This will make it easier later when reading messages from it on your channel.

Once your bot is created, it will generate a token for you in order to communicate to Slack API.

Do not forget to add the bot user to your channel!

Spotify

The second step is to register an APP in your Spotify account. You can follow the instructions on how to do this in [here](#)

Configuration

Once you have all the details, you are going to need to create a configuration file so that the CLI can get all the parameters.

By default, the program will look into a hidden file named `.slacksound` under your home directory but you can also specify the location manually.

```
slacksound --configuration /Users/slack/myconfig.cfg
```

The configuration must be like as per below:

```
[spotify]
username = <username>
password = <password>
client_id = <client_id>
client_secret = <client_secret>
playlist = <playlist_name>
callback_url = <callback_url>
scope = <scope>

[slack]
token = <bot_token>
channel = <channel_name>
reaction = <emoji_reaction>
count = <number_of_reaction_counts>
```

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

Submit Feedback

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.

Get Started!

Ready to contribute? Here's how to set up *slacksound* for local development.

1. Clone your fork locally:

```
$ git clone https://github.com/wefner/slacksound
```

2. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your clone for local development:

```
$ mkvirtualenv slacksound  
$ cd slacksound/  
$ python setup.py develop
```

3. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

4. Commit your changes and push your branch to the server:

```
$ git add .  
$ git commit -m "Your detailed description of your changes."  
$ git push origin name-of-your-bugfix-or-feature
```

5. Submit a merge request

slacksound package

Submodules

slacksound.slackapi module

Main code for slackapi

class `slacksound.slackapi.Attachment` (*attachment_details*)

Bases: `object`

Model for an Attachment

Not all attributes are populated here though.

<https://api.slack.com/docs/message-attachments>

author_link

A valid URL that will hyperlink the `author_name` text mentioned above.

Returns: `string`

author_name

Small text used to display the author's name.

Returns: `string`

title

Title of the attachment

Returns: `string`

class `slacksound.slackapi.Channel` (*slack_instance, channel_details*)

Bases: `object`

Model for a Channel

Not all attributes are populated here though.

<https://api.slack.com/types/channel>

channel_id

Channel ID

Returns: string

created

Unix time converted to datetime object

Returns: datetime object

history

Chat history of the channel

Returns: list of Message objects

is_general

Whether the channel is general or not

Returns: boolean

name

Name of the channel

Returns: string

name_normalized

Normalized name of the channel

Returns: string

class `slacksound.slackapi.Group`(*slack_instance*, *group_details*)

Bases: object

Model for a group

Not all attributes are populated here though.

<https://api.slack.com/types/group>

created

Unix time converted to datetime object

Returns: datetime object

group_id

Group ID

Returns: string

history

Chat history of the group

Returns: list of Message objects

is_general

Whether the group is general or not

Returns: boolean

name

Name of the group

Returns: string

name_normalized

Normalized name of the group

Returns: string

class `slacksound.slackapi.Member` (*member_details*)

Bases: object

Model for a Member

Not all attributes are populated here though.

<https://api.slack.com/types/user>

color

Used in some clients to display a colored username.

Returns: string

deleted

Active/Inactive users

Returns: boolean

email

Email of the member

Returns: string

is_admin

Whether this user is admin or not

Returns: boolean

member_id

Member ID

Returns: string

class `slacksound.slackapi.Message` (*message_details*)

Bases: object

Model for a Message

Not all attributes are populated here though.

<https://api.slack.com/events/message>

attachments

Attachments for a Message, if any

Returns: list of Attachment objects

datetime

Unix time converted to datetime object

Returns: datetime object

reaction

Reactions of the message

Returns: list of Reaction objects

text

Text of the message

Returns: string

type
SubType of the Message
Returns: string

unix_time
Unix time. Useful for time comparison
Returns: float

user
ID of the user speaking or sent the message
Returns: string

class `slacksound.slackapi.Reaction` (*reaction_details*)
Bases: object
Model for a Reaction
These are Emoji icons for a message

count
Count of a reaction
Returns: integer

name
Name of the reaction
Returns: string

users
User IDs who reacted on the reaction
Returns: list of user ID

class `slacksound.slackapi.Slack` (*token*, *bot=False*)
Bases: object
SlackClient Wrapper

channels
Gets all channels in a Slack team.
Parameters ****kwargs** – extra kwargs
Returns: list of Channel objects

get_channel_by_name (*channel_name*)
Gets one channel in a Team
Parameters **channel_name** – string
Returns: Channel object

get_group_by_name (*group_name*)
Gets one group in a Team
Parameters **group_name** – string
Returns: Group object

groups
Gets all groups in a Slack team.
Parameters ****kwargs** – extra kwargs

Returns: list of Group objects

post_message (*message*, *channel*)

Posts a message in a group or channel as the user who is owner of the token

Parameters

- **message** – string
- **channel** – string

Returns:

users

Gets all users in a Slack team.

Parameters ****kwargs** – extra kwargs

Returns: list of Member objects

slacksound.slacksound module

Main code for slacksound

slacksound.slacksound.**SlackSound**

alias of Config

slacksound.slacksound.**connect_spotify** (*credentials*)

slacksound.slacksound.**get_arguments** ()

This get us the cli arguments. Returns the args as parsed from the argparser.

slacksound.slacksound.**get_config_details** (*credentials*)

slacksound.slacksound.**get_credentials** (*filename=False*)

Parameters **filename** – path of the filename

Reads credentials file

Locates credentials file in ~/.slacksound and loads them by default.

This file must be a INI type and if a filename is provided, it will try to load this one.

Returns ConfigParser instance

slacksound.slacksound.**get_most_popular_track** (*tracks*)

slacksound.slacksound.**main** ()

Main method. This method holds what you want to execute when the script is run on command line.

slacksound.slacksound.**sanitize_title** (*title*)

Gets the artist and song name only

Splits the string by the the first open parenthesis and gets first item in the list

Examples

in: ‘Eric Clapton - Cocaine (Original Video)’ out: ‘Eric Clapton - Cocaine’

Parameters **title** – string

Returns: string

`slacksound.slacksound.setup_logging(args)`

This sets up the logging. Needs the args to get the log level supplied :param args: The arguments returned gathered from argparse

slacksound.slacksoundexceptions module

Custom exception code for slacksound

slacksound.spotifyclient module

Main code for spotifyclient

class `slacksound.spotifyclient.Playlist(username, spotify_instance, playlist_details)`

Bases: object

Playlist model

add_track (*track_id*)

Add a track to the playlist

Parameters *track_id* – string

Returns: Boolean

collaborative

Whether it is collaborative or not

Returns: boolean

delete_all_tracks ()

Deletes all tracks that are in the playlist

Returns: Snapshot ID

href

Link of the Playlist

Returns: string

name

Name of the Playlist Returns:

playlist_id

Playlist ID

Returns: string

public

Whether it is a public playlist or not

Returns: boolean

tracks

Get all tracks in the playlist

Returns: list of Track objects

uri

URI of the playlist

Returns: string

class `slacksound.spotifyclient.SpotifyClient` (*client_id, client_secret, username, password, callback, scope*)

Bases: `object`

get_playlist_by_name (*playlist_name*)

Looks into all playlists and returns the one that matched

Parameters `playlist_name` – string

Returns: Playlist object

get_track_by_title (*track_title, limit=5*)

Looks up on Spotify for a text string and returns tracks if found

The limit is an optional argument to retrieve more results

Examples

‘Eric Clapton - Cocaine’ ‘Bon Jovi - Livin’ On A Prayer’

Parameters

- **track_title** – string
- **limit** – integer

Returns: list of Track objects

playlists

Get all playlists for the user within its scope

It also passes username and Spotify instance as it is needed for Playlist object

Returns: list of Playlist objects

class `slacksound.spotifyclient.Track` (*track_details*)

Bases: `object`

Model for a Track

duration_ms

Duration in milliseconds

Returns: integer

name

Name of the track

Returns: string

popularity

Popularity of the track

Returns: integer

track_id

Track ID

Returns: string

uri

URI of the track

Returns: URI

Module contents

slacksound package

Import all parts from slacksound here

CHAPTER 6

Credits

Development Lead

- Oriol Fabregas <fabregas.oriol@gmail.com>

Contributors

None yet. Why not be the first?

CHAPTER 7

History

CHAPTER 8

0.0.1 (2017-10-13)

- First release
- Methods for Spotify API
- Slackclient wrapper

CHAPTER 9

0.0.2 (2017-10-14)

- CLI draft
- Credentials parsing
- Minimal logic

CHAPTER 10

0.0.3 (2017-10-14)

- Reaction name and reaction count configurable

CHAPTER 11

0.0.4 (2017-10-26)

- Refactored SpotifyClient with objects and methods
- Refactored SlackSound interface

CHAPTER 12

0.0.5 (2017-10-27)

- Entrypoint for main

CHAPTER 13

0.1.0 (2017-10-27)

- Try a private channel first or fallback as a public one if fails

CHAPTER 14

0.1.1 (2017-10-27)

- Logging for user

CHAPTER 15

Indices and tables

- `genindex`
- `modindex`
- `search`
- `search`

S

`slacksound`, [18](#)
`slacksound.slackapi`, [11](#)
`slacksound.slacksound`, [15](#)
`slacksound.slacksoundexceptions`, [16](#)
`slacksound.spotifyclient`, [16](#)

A

add_track() (slacksound.spotifyclient.Playlist method), 16
Attachment (class in slacksound.slackapi), 11
attachments (slacksound.slackapi.Message attribute), 13
author_link (slacksound.slackapi.Attachment attribute), 11
author_name (slacksound.slackapi.Attachment attribute), 11

C

Channel (class in slacksound.slackapi), 11
channel_id (slacksound.slackapi.Channel attribute), 12
channels (slacksound.slackapi.Slack attribute), 14
collaborative (slacksound.spotifyclient.Playlist attribute), 16
color (slacksound.slackapi.Member attribute), 13
connect_spotify() (in module slacksound.slacksound), 15
count (slacksound.slackapi.Reaction attribute), 14
created (slacksound.slackapi.Channel attribute), 12
created (slacksound.slackapi.Group attribute), 12

D

datetime (slacksound.slackapi.Message attribute), 13
delete_all_tracks() (slacksound.spotifyclient.Playlist method), 16
deleted (slacksound.slackapi.Member attribute), 13
duration_ms (slacksound.spotifyclient.Track attribute), 17

E

email (slacksound.slackapi.Member attribute), 13

G

get_arguments() (in module slacksound.slacksound), 15
get_channel_by_name() (slacksound.slackapi.Slack method), 14
get_config_details() (in module slacksound.slacksound), 15

get_credentials() (in module slacksound.slacksound), 15
get_group_by_name() (slacksound.slackapi.Slack method), 14
get_most_popular_track() (in module slacksound.slacksound), 15
get_playlist_by_name() (slack-sound.spotifyclient.SpotifyClient method), 17
get_track_by_title() (slack-sound.spotifyclient.SpotifyClient method), 17

Group (class in slacksound.slackapi), 12
group_id (slacksound.slackapi.Group attribute), 12
groups (slacksound.slackapi.Slack attribute), 14

H

history (slacksound.slackapi.Channel attribute), 12
history (slacksound.slackapi.Group attribute), 12
href (slacksound.spotifyclient.Playlist attribute), 16

I

is_admin (slacksound.slackapi.Member attribute), 13
is_general (slacksound.slackapi.Channel attribute), 12
is_general (slacksound.slackapi.Group attribute), 12

M

main() (in module slacksound.slacksound), 15
Member (class in slacksound.slackapi), 13
member_id (slacksound.slackapi.Member attribute), 13
Message (class in slacksound.slackapi), 13

N

name (slacksound.slackapi.Channel attribute), 12
name (slacksound.slackapi.Group attribute), 12
name (slacksound.slackapi.Reaction attribute), 14
name (slacksound.spotifyclient.Playlist attribute), 16
name (slacksound.spotifyclient.Track attribute), 17
name_normalized (slacksound.slackapi.Channel attribute), 12

name_normalized (slacksound.slackapi.Group attribute),
12

P

Playlist (class in soundsound.spotifyclient), 16
playlist_id (soundsound.spotifyclient.Playlist attribute), 16
playlists (soundsound.spotifyclient.SpotifyClient at-
tribute), 17
popularity (soundsound.spotifyclient.Track attribute), 17
post_message() (soundsound.slackapi.Slack method), 15
public (soundsound.spotifyclient.Playlist attribute), 16

R

Reaction (class in soundsound.slackapi), 14
reaction (soundsound.slackapi.Message attribute), 13

S

sanitize_title() (in module soundsound.soundsound), 15
setup_logging() (in module soundsound.soundsound), 15
Slack (class in soundsound.slackapi), 14
SlackSound (in module soundsound.soundsound), 15
soundsound (module), 18
soundsound.slackapi (module), 11
soundsound.soundsound (module), 15
soundsound.soundsoundexceptions (module), 16
soundsound.spotifyclient (module), 16
SpotifyClient (class in soundsound.spotifyclient), 16

T

text (soundsound.slackapi.Message attribute), 13
title (soundsound.slackapi.Attachment attribute), 11
Track (class in soundsound.spotifyclient), 17
track_id (soundsound.spotifyclient.Track attribute), 17
tracks (soundsound.spotifyclient.Playlist attribute), 16
type (soundsound.slackapi.Message attribute), 13

U

unix_time (soundsound.slackapi.Message attribute), 14
uri (soundsound.spotifyclient.Playlist attribute), 16
uri (soundsound.spotifyclient.Track attribute), 17
user (soundsound.slackapi.Message attribute), 14
users (soundsound.slackapi.Reaction attribute), 14
users (soundsound.slackapi.Slack attribute), 15