Local Music Bot Documentation

Release 1.0.1

Natsulus

Jul 22, 2017

Getting Started

1 Installing 3

2 Config 5

3 Commands 7

4 Query 15

5 Role 17

6 Perms 21

7 Library 25

8 TODO 27

9 Changelog 29

The Local Music Bot is a Discord Bot for Local Music with library and role management.

If you have any feedback or problems, feel free to add `Natsulus#5647` on Discord or submit an issue on Github or join the Support Server.

If you wish to contribute and know how to code with Node.js, you can make a pull request on the Dev branch of the Github Repository.

Note: The bot and docs are still a work-in-progress.

CHAPTER 1

Installing

Requirements

Windows

- NodeJS - Requires v5.0.0 or above due to ES6 code.
- Python 2.7 - Requires v2.7.x, not v3.x.x.
- Visual Studio - Select Custom Installation and make sure Visual C++ under Programming Languages is selected.
  - Your Visual Studio installation ideally has to be recent.
  - Can be Express, Community or Enterprise Edition, but NOT `VS Code`.
- FFmpeg - Add bin folder to PATH Environment Variable.

Mac

- NodeJS - Requires v5.0.0 or above due to ES6 code.
- Python 2.7 - Requires v2.7.x, not v3.x.x. Should be already installed by default.
- Xcode Command Line Tools

```
$ xcode-select --install
```

- FFmpeg

```
$ brew install ffmpeg
```

Linux (Debian-based)

- NodeJS Linux - Requires v5.0.0 or above due to ES6 code.

- Python 2.7 - Requires v2.7.x, not v3.x.x. Should be already installed by default.
- Build Essential

```
$ sudo apt-get install build-essential
```

- FFmpeg

```
$ sudo apt-get install ffmpeg
```

If on Ubuntu 14.04, use this instead:

```
$ sudo add-apt-repository ppa:mc3man/trusty-media && sudo apt-get update && sudo apt-
↪get install ffmpeg
```

Installation

1. Download the repository. (You can use git clone or the Download ZIP button Github provides.)
2. Open the *config.json* file and edit it.
3. Open a Command Prompt/Terminal at the directory of the downloaded repository.
4. Run `npm install` to install the dependencies.
5. Run `npm start` to start up the bot.
6. Start using the bot!

CHAPTER 2

Config

config.json

prefix The prefix the bot uses.

owner The owner’s Discord ID.

token Your bot’s token. Defaults to email and password if not provided.

email Your bot’s email. Optional if token provided.

password Your bot’s password. Optional if token provided.

allowedChannels Array of **text** channel IDs which the bot will listen to. Used to avoid clogging up channel logs.

Example

```
{
    "prefix": ">",
    "owner": "345678909876543",
    "token": "ABC123def987-zxyNOP546",
    "email": "",
    "password": "",
    "allowedChannels": ["23456789098765432", "23456789098765432", "23456789098765432"]
}
```

CHAPTER 3

Commands

Query

search *Query*

Perm: *query.search*
Aliases: find

Searches for songs based on a *Query* and returns a song or list.

queue *Query*

Perm: *query.queue*
Aliases: insert, q

Searches for songs based on a *Query* and queues a song or list.

Voice

join *ChannelName*

Perm: *voice.join*
Aliases:

Joins a voice channel if not already in one. ChannelName is case-insensitive and provides partial matching.

leave

Perm: *voice.leave*
Aliases:

Leaves a voice channel if connected to one.

View

current

Perm: *view.current*
Aliases: playing, song

Provides detail about the song currently playing.

list

Perm: *view.list*
Aliases: songs

Lists all songs in queue as well as the song currently playing.

cover

Perm: *view.cover*
Aliases: art, artwork, albumart, image

Get the album art of the current song if available.

Queue

autoplay *on* | *off* | *status*

Perm: *role.mod.add*
Aliases: auto, ap

Enables or disables autoplay.

unqueue *NumberOrRange*

Perm: *queue.remove*
Aliases: remove

Removes the track/s from the queue at the provided Number/Range.

shuffle

Perm: *queue.shuffle*
Aliases: randomise, randomize

Shuffles the queue.

skip

Perm: *queue.skip.novote*
Aliases: next

Skips current song.

skip vote

Perm: *queue.skip.vote*
Aliases: next

Calls a vote to skip a song if no vote has started, else adds to vote to skip song.

Role

role

Perm: *role.view.own*
Aliases: r

View details about own role.

role *User#Tag*

Perm: *role.view.other*
Aliases: r

View details about the role of User#Tag.

role view *ID*

Perm: *role.view.detail*
Aliases: r

View details about specified role.

role list

Perm: *role.view.list*
Aliases: r

Lists all roles with their ID and Name.

role add *ID* | *Name* | *Description*

Perm: *role.mod.add*
Aliases: r

Adds a new role. Description is optional.

role delete *ID*

Perm: *role.mod.delete*
Aliases: r

Deletes the role with the specified role. All users with the deleted role will have their role set to the default role.

role edit *ID* | *Key* | *Value*

Perm: *role.mod.edit*
Aliases: r

Edits the key for the specified role with the value. Cannot edit the keys Perm or Default.

role set *ID* *User#Tag*

Perm: *role.mod.set*
Aliases: r

Sets the role of User#Tag with the specified role. Can enter multiple User#Tag, separated by `" | "`. e.g. `>role set 5 Bob#1234 | Fred#0420 | Tim#000`

role default

Perm: *role.default.view*
Aliases: r

View details on the default role.

role default *ID*

Perm: *role.default.set*
Aliases: r

Sets the default role to the specified role. ID must be greater than 5.

role give *ID Perm*

Perm: *role.perm.give*
Aliases: r

Gives the specified role the specified Perm. Can enter multiple perm, separated by a space. e.g. `>role give 5 role.default.view role.view playlist`

role remove *ID Perm*

Perm: *role.perm.remove*
Aliases: r

Removes the specified Perm from the specified role. Can enter multiple perm, separated by a space. e.g. `>role remove 5 role.default.view role.view playlist`

Library

library

Perm: *library.view.active*
Aliases: lib

View details of the active library.

library *Name*

Perm: *library.view.detail*
Aliases: lib

View details of specified library.

library list

Perm: *library.view.list*
Aliases: lib

View list of libraries.

library add *Name* | *Path* | *Description*

Perm: *library.mod.add*
Aliases: lib

Add a new library. Description is optional.

library delete *Name*

Perm: *library.mod.delete*
Aliases: lib

Deletes the library with the specified Name.

library edit *Name* | *Key* | *Value*

Perm: *library.mod.edit*
Aliases: lib

Edits the key for the library with the specified Name with the value. Cannot edit the key Active.

library select *Name*

Perm: *library.mod.select*
Aliases: lib

Selects the specified library as the active library.

update

Perm: *library.update.active*
Aliases:

Check active library for any changes and updates the library.

update *LibraryName*

Perm: *library.update.other*
Aliases:

Check specified library for any changes and updates the library.

Playlist

playlist *Name*

Perm: *playlist.view.detail*
Aliases: pl

Lists all songs in the specified playlist.

playlist list

Perm: *playlist.view.list*
Aliases: pl

Lists all playlist names.

playlist save *Name*

Perm: *playlist.save*
Aliases: pl

Save the current song and queue as a playlist with the specified name.

playlist load *Name*

Perm: *playlist.load*
Aliases: pl

Load the specified playlist into the queue.

playlist delete *Name*

Perm: *playlist.delete.own* | *playlist.delete.other*
Aliases: pl

Deletes the specified playlist if you created it and have *playlist.delete.own* perm.

Deletes the specified playlist if you didn’t create it and have *playlist.delete.other* perm.

CHAPTER 4

Query

Options

Query	Details
title	Case-Insensitive & Partial Matching
album	Case-Insensitive & Partial Matching
artist	Case-Insensitive & Partial Matching
allbumartist	Case-Insensitive & Partial Matching
filename	Case-Insensitive & Partial Matching
duration	Number for amount of seconds. Negative for *shorter than*, Positive for *longer than*.
page	Number. Results are limited to 20 per page.
track	Number or Range. This is the number seen in the results and not the track number of an album.
sortby	none, title, album, artist, albumartist, filename, duration. Sorts in ascending order, add - in front for descending.
random	Provide any value to get a random song based on other query values except page, track and sortby.

Examples

Search for songs with artist SomeArtist, album SomeAlbum, getting results 21-40 (Page 2) when sorted by title then album in ascending order,then artist in descending order:

```
>search artist:  SomeArtist album:  SomeAlbum page:  2 sortby:  title album -artist
```

Search for songs with album SomeAlbum with a duration shorter than 420 seconds:

```
>search album:  SomeAlbum duration:  -420
```

Search for songs with album SongAlbum and select track #5:

```
>search album:  SomeAlbum track:  5
```

Search for songs with artist Space Artist and select track 3-8 when not sorted:

```
>search artist:  Space Artist track:  3-8 sortby:  none
```

Search for a random song from songs with album SongAlbum:

```
>search album:  SomeAlbum random:  Anything Can Be Put For Random's Value But
There Must Be Something
```

Search for a random song from the library:

```
>search random:  asdf
```

Notes

- If no sortby value is provided it defaults to the sortby value: `album artist albumartist title filename`

CHAPTER 5

Role

Keys

ID

The numbered position of the role in the hiearchy. The lower the number, the higher the hierarchical position with #0 reserved for the Owner.
This must be unique.

Name

Name of the Role.
This must be unique.

Description

A description of the role.

Perm

An array/list of *Perms* the role has.

Default

Only one role has the value of this key as true at one time.
Role with this key as true is the default role a new user gets assigned to.

Provided Roles

The following roles come with the bot:

Owner

ID: 0
Description: The Owner of the bot.
Perm: Code bypasses perm requirements.
Default: false

Partner

ID: 1
Description: A user the Owner can absolutely trust.
Perm: query, voice, view, queue, role, library.view, library.mod.select, library.update, playlist
Default: false

Admin

ID: 3
Description: A user who moderates other users.
Perm: query, voice, view, queue, role.view, role.mod, role.default.view, library.view, library.mod.select, library.update, playlist
Default: false

Moderator

ID: 5
Description: A user who assists with moderating other users.
Perm: query, voice, view, queue, role.view, role.mod.set, library.view.active, library.update.data.active, playlist.view, playlist.save, playlist.load, playlist.delete.own
Default: false

Listener

ID: 8
Description: A user who uses the bot.
Perm: query, voice, view, queue.shuffle, queue.skip.vote, role.view.own, playlist.view, playlist.save, playlist.load, playlist.delete.own
Default: false

Member

ID: 10
Description: A normal user.
Perm: query.search, view.current, view.list
Default: true

CHAPTER 6

Perms

Perm List

- query
  - search
  - queue
- voice
  - join
  - leave
- view
  - current
  - list
  - cover
- queue
  - remove
  - shuffle
  - skip
    - novote
    - vote
- role
  - view
    - own
    - other

        - detail
        - list
    - mod
        - add
        - delete
        - edit
        - set
    - default
        - view
        - set
    - perm
        - give
        - remove
- library
    - view
        - active
        - detail
        - list
    - mod
        - add
        - delete
        - edit
        - select
    - update
        - active
        - other
- playlist
    - view
        - detail
        - list
    - save
    - load
    - delete
        - own
        - other

Notes

Use dot notation to select sub perms. Selecting main perm will select all sub perms.

e.g. `voice = voice.join & voice.leave`

CHAPTER 7

Library

Keys

Name

The name of the library.
This must be unique.

Path

The root path of the library. Paths should not use and should instead use /.
This must be unique.

Description

A description of the library.

Active

Only one library has the value of this key as true at one time.
Library with this key as true is the library that is currently being used.

Notes

- New libraries will take a while to load depending on the size of the library and the disk's read speed as they must open each file to get their metadata.

- Libraries cannot detect files that were modified while the library was not active.

CHAPTER 8

TODO

v1

- Prettify the messages.
- Support @mention
- Support new members joining server.

v2 (Undecided)

- Change Bot Name to 'Music Library Bot'.
- Implement Streaming Support (Youtube, SoundCloud, etc).
- Support modified files when bot is offline.

v3 (Undecided)

- Stream Site DL to File in Library?

CHAPTER 9

Changelog

1.0.1

- Fixed Perm Checking
- Fixed Role Set Command
- Started Planning v1.1 Update
- Created a Support Server

1.0.0

- Version 1.0.0 of the bot is now released!
- All commands are usable.
- Role Management is available with perms.
- Library Management & File Monitoring while online.
- Updates Library for all offline changes except for modified/replaced files.