pycmus Documentation

Matthew Treinish

May 29, 2018

Contents

Contents:

CHAPTER 1

pycmus

A python library for sending commands to the cmus music player:

https://cmus.github.io/

It uses the same socket interface as the cmus-remote command.

Complete documentation is here: http://pycmus.readthedocs.io/en/latest/

1.1 Usage

Using pycmus is pretty straightforward you just need to init a PyCmus object and then issue commands to it. For example:

```
from pycmus import remote

cmus = remote.PyCmus()
print(cmus.status())
```

will connect to a running cmus instance (with the socket file in the default location) and print the player status.

For a complete API documentation see: *PyCmus API*.

CHAPTER 2

PyCmus API

class pycmus.remote.**PyCmus**(*server=None*, *socket_path=None*, *password=None*, *port=3000*)

Bases: object

PyCmus remote class

This class is used to create a PyCmus remote object that is used to send commands to a running cmus. It can be used to connect to either a locally running cmus or a cmus on a remote machine that is configured to listen over the network. If neither a server or a socket file are provided the PyCmus object will look for a running cmus in the default locations and try to connect to that.

Parameters

- **server** (*str*) – The remote host to connect to the cmus socket on
- **socket_path** (*str*) – The path to the local unix socket for cmus
- **password** (*str*) – The password to use when establishing a remote connection. It is a required field if a server is provided. If a socket_path is used this is ignored
- **port** (*int*) – The port to use for remote connections. If one is not provided it will just use the default port of 3000.

get_status_dict()

Send a status command and format response as a dictionary

Return status The player status, it is a newline seperated string with the current state of the player.

Return type dict

player_next()

Send a player next command.

player_pause()

Send a player pause command.

player_pause_playback()

Send a player pause playback command.

player_play()
Send a player play command.

player_play_file(*play_file*)
Send a player play command with a file

Parameters **play_file** (*str*) – The path or url to the file to play

player_prev()
Send a player previous command.

player_stop()
Send a player stop command.

seek(*seek*)
Send a player seek command

Parameters **seek** – The position to seek the player to. This can either be a raw integer which will be the position in number of secs (where 0 is the start of the file) or it can be an +/- # offset where the position will either either move forward or backwards respectively the number of seconds specified

send_cmd(*cmd*)
Send a raw command to cmus

Parameters **cmd** (*str*) – The command to send to cmus

Return resp The response from cmus from the issued command

Return type str

set_volume(*volume*)
Send a player set volume command

Parameters **volume** (*int*) – the volume to set the volume to

status()
Send a status command

Return status The player status, it is a newline seperated string with the current state of the player.

Return type str

toggle_repeat()
Send a toggle repeat command.

toggle_shuffle()
Send a toggle shuffle command.

Python Module Index

p

Index

G

P

S

T