

---

# **pymus Documentation**

**Matthew Treinish**

**May 11, 2018**



---

## Contents

---

<b>1</b>	<b>pymus</b>	<b>3</b>
1.1	Usage . . . . .	3
<b>2</b>	<b>PyCmus API</b>	<b>5</b>
	<b>Python Module Index</b>	<b>7</b>



Contents:



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 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.

**p**

`pymus.remote`, 5



### G

`get_status_dict()` (`pymus.remote.PyCmus` method), 5

### P

`player_next()` (`pymus.remote.PyCmus` method), 5

`player_pause()` (`pymus.remote.PyCmus` method), 5

`player_pause_playback()` (`pymus.remote.PyCmus` method), 5

`player_play()` (`pymus.remote.PyCmus` method), 5

`player_play_file()` (`pymus.remote.PyCmus` method), 6

`player_prev()` (`pymus.remote.PyCmus` method), 6

`player_stop()` (`pymus.remote.PyCmus` method), 6

`PyCmus` (class in `pymus.remote`), 5

`pymus.remote` (module), 5

### S

`seek()` (`pymus.remote.PyCmus` method), 6

`send_cmd()` (`pymus.remote.PyCmus` method), 6

`set_volume()` (`pymus.remote.PyCmus` method), 6

`status()` (`pymus.remote.PyCmus` method), 6

### T

`toggle_repeat()` (`pymus.remote.PyCmus` method), 6

`toggle_shuffle()` (`pymus.remote.PyCmus` method), 6