

---

# **SnakeJazz**

***Release 0.1.1***

**Martin Chalela**

**Dec 06, 2020**



**CONTENTS:**

<b>1</b>	<b>Repository and Issues</b>	<b>3</b>
1.1	API Index . . . . .	3
1.2	Requirements . . . . .	6
1.3	Installation . . . . .	6
1.4	Licence . . . . .	6
<b>2</b>	<b>Indices and tables</b>	<b>7</b>
	<b>Python Module Index</b>	<b>9</b>
	<b>Index</b>	<b>11</b>



**SnakeJazz** provides decorators to let you listen to the running status of your Python functions. Most definitely inspired by Rick and Morty ;)

These are the decorators:

- ***snakejazz.zzz***: You can choose to reproduce a sound at the moment your function starts to execute, when it finishes or when an error occurs. A different sound for each event can be given.
- ***snakejazz.www***: Exactly the same as zzz, but you can specify youtube links and the audio will be downloaded.
- ***snakejazz.rattle***: Rattle from start to finish. This will loop the sound until your function ends. You can either give a local path or a youtube link.

All three of them can be used directly to run with the default configuration. But you can also give some custom sounds for a more enjoyable moment.

#### **Author**

Martin Chalela (E-mail: [tinchochalela@gmail.com](mailto:tinchochalela@gmail.com))



## REPOSITORY AND ISSUES

<https://github.com/mchalela/SnakeJazz>

### 1.1 API Index

These are the decorators:

- *snakejazz.zzz*:

You can choose to reproduce a sound at the moment your function starts to execute, when it finishes or when an error occurs. A different sound for each event can be given.

- *snakejazz.www*:

Exactly the same as *zzz*, but you can specify youtube links and the audio will be downloaded.

- *snakejazz.rattle*:

Rattle from start to finish. This will loop the sound until your function ends. You can either give a local path or a youtube link.

---

#### 1.1.1 Module `snakejazz.snakejazz`

SnakeJazz.

Listen to the running status of your `Snake` Python functions.

**exception** `snakejazz.snakejazz.SnakeNotFoundError`

Bases: `FileNotFoundError`

Raised when the file can't be found.

**exception** `snakejazz.snakejazz.URLError`

Bases: `OSError`

Raised when the url is invalid.

`snakejazz.snakejazz.play_sound(sound_path, loops=0)`

Reproduce the sound.

The library PyGame is used to reproduce sounds.

**Parameters**

- **sound\_path** (*string*, *path*) – Path to the sound file.
- **loops** (*int*) – Number of times the sound will be played.  
0: A single time  
-1: Infinite loop

```
snakejazz.snakejazz.get_sound(yt_url=None, yt_id='ahgcD1xjRiQ', use_cache=True)
```

Download the sound.

The library youtube-dl is used to download sounds.

**Parameters**

- **yt\_url** (*string*, *link*) – Youtube link. The audio will be extracted from the video.
- **yt\_id** (*str*, *id*) – Youtube video id. If this is given the full url will be completed as [https://www.youtube.com/watch?v=yt\\_id](https://www.youtube.com/watch?v=yt_id)
- **use\_cache** (*bool*) – When True, a sound will be downloaded just once and save it for later use if needed.

```
snakejazz.snakejazz.zzz(method=None, *, when_start=False, when_finish=True,  
                        when_error=False)
```

Sound decorator to notify the execution status of a function.

**Parameters**

- **method** (*callable*) – Function, class method or any callable object. SnakeJazz will track the starting and finishing event and play the desired sound.
- **when\_start** (*string*, *path*, *optional*) – Path to the sound file that will be played at the same instant that the execution of ‘method’ starts. A new process handles the reproduction of the sound.
- **when\_finish** (*string*, *path*, *optional*) – Path to the sound file that will be played at the same instant that the execution of ‘method’ ends. A new process handles the reproduction of the sound.
- **when\_error** (*string*, *path*, *optional*) – Path to the sound file that will be played if an exception occurs during the execution of ‘method’. If an error occurs, no finishing sound is played. A new process handles the reproduction of the sound.

**Notes**

SnakeJazz uses PyGame API to reproduce sounds. The default sounds distributed with SnakeJazz belong to the respective creators.

Rhodesmas:

>> Downloaded from <https://freesound.org/people/rhodesmas/packs/17958/>

```
snakejazz.snakejazz.www(method=None, *, when_start=False, when_finish=True,  
                        when_error=False)
```

Sound decorator to notify the execution status of a function.

**Parameters**

- **method** (*callable*) – Function, class method or any callable object. SnakeJazz will track the starting and finishing event and play the desired sound.



- **when\_start** (*string, link, optional*) – Youtube link to the audio that will be played at the same instant that the execution of ‘method’ starts. A new process handles the reproduction of the sound.
- **when\_finish** (*string, link, optional*) – Youtube link to the audio that will be played at the same instant that the execution of ‘method’ ends. A new process handles the reproduction of the sound.
- **when\_error** (*string, link, optional*) – Youtube link to the audio that will be played if an exception occurs during the execution of ‘method’. If an error occurs, no finishing sound is played. A new process handles the reproduction of the sound.

## Notes

SnakeJazz uses PyGame API to reproduce sounds and YoutubeDL to download audio from youtube videos. The default sounds distributed with SnakeJazz belong to the respective creators.

Rhodesmas:

>> Downloaded from <https://freesound.org/people/rhodesmas/packs/17958/>

```
snakejazz.snakejazz.rattle (method=None, *, zound=None, url='https://youtu.be/ahgcD1xjRiQ')
```

Reproduce the sound in loop until the execution is completed.

## Parameters

- **method** (*callable*) – Function, class method or any callable object. SnakeJazz will track the strating and finishing event and play the desired sound.
- **zound** (*string, path, optional*) – Path to the sound file that will be played during the execution of ‘method’. A new process handles the reproduction of the sound.
- **url** (*string, path, optional*) – Youtube link to the audio that will be played during the execution of ‘method’. A new process handles the reproduction of the sound.

## Notes

SnakeJazz uses PyGame API to reproduce sounds and YoutubeDL to download audio from youtube videos. The default sounds distributed with SnakeJazz belong to the respective creators.

Rhodesmas:

>> Downloaded from <https://freesound.org/people/rhodesmas/packs/17958/>

## 1.1.2 Package snakejazz.sounds

Collection of sounds to use with SnakeJazz.

```
snakejazz.sounds.RICK_AND_MORTY = 'https://youtu.be/ahgcD1xjRiQ'
```

Youtube link to original Snake Jazz song by Rick and Morty.

**Type** str, url

```
snakejazz.sounds.RHODESMAS
```

Paths to all sounds available in this pack.

**Type** dict

## 1.2 Requirements

You will need to install ffmpeg for audio processing. For example, in Ubuntu/Debian:

```
$ sudo apt update $ sudo apt install ffmpeg
```

You will also need a nice set of headphones to run SnakeJazz.

## 1.3 Installation

The easiest way to install is using pip:

```
$ pip install snakejazz
```

This will install the latest stable version on PyPI.

If you want to use the latest development version from github, unpack or clone the *repo* <<https://github.com/mchalela/SnakeJazz>> on your local machine, change the directory to where setup.py is, and install using setuptools:

```
$ python setup.py install
```

or pip:

```
$ pip install -e .
```

## 1.4 Licence

### MIT License

Copyright (c) 2020 Martin Chalela

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## INDICES AND TABLES

- `genindex`
- `search`



## PYTHON MODULE INDEX

### S

`snakejazz.snakejazz`, [3](#)

`snakejazz.sounds`, [5](#)



## INDEX

### G

`get_sound()` (in module *snakejazz.snakejazz*), 4

### M

module

*snakejazz.snakejazz*, 3

*snakejazz.sounds*, 5

### P

`play_sound()` (in module *snakejazz.snakejazz*), 3

### R

`rattle()` (in module *snakejazz.snakejazz*), 5

*RHODESMAS* (in module *snakejazz.sounds*), 5

*RICK\_AND\_MORTY* (in module *snakejazz.sounds*), 5

### S

*snakejazz.snakejazz*

    module, 3

*snakejazz.sounds*

    module, 5

*SnakeNotFoundError*, 3

### U

*URLError*, 3

### W

`www()` (in module *snakejazz.snakejazz*), 4

### Z

`zzz()` (in module *snakejazz.snakejazz*), 4