

---

# **PIFX Documentation**

***Release 0.0.2***

**Chaoyi Zha**

**May 28, 2020**



---

## Contents

---

|                            |          |
|----------------------------|----------|
| <b>Python Module Index</b> | <b>5</b> |
| <b>Index</b>               | <b>7</b> |



To use PIFX, initiate an instance of the PIFX class with your API token to use its functions.

```
import pifx

p = pifx.PIFX(api_key='API_KEY_GOES_HERE')

p.toggle_power() # toggle all lights
p.toggle_power('label:Bedroom') # toggle light with label "Master Bedroom"
p.set_state(color='blue', brightness='0.85') # set brightness to 85% and color to blue
p.pulse_lights(color='red', duration=2.5) # pulse lights for 2.5 seconds
```

PIFX Usage Documentation:

**class** pifx.PIFX(*api\_key*, *http\_endpoint=None*)

Main PIFX class

**activate\_scene** (*scene\_uuid*, *duration=1.0*)

Activate a scene.

See <http://api.developer.lifx.com/docs/activate-scene>

**scene\_uuid: required String** The UUID for the scene you wish to activate

**duration: Double** The time in seconds to spend performing the scene transition. default: 1.0

**breathe\_lights** (*color*, *selector='all'*, *from\_color=None*, *period=1.0*, *cycles=1.0*, *persist=False*,  
*power\_on=True*, *peak=0.5*)

Perform breathe effect on lights.

**selector: String** The selector to limit which lights will run the effect. default: all

**color: required String** Color attributes to use during effect. See `set_state` for more.

**from\_color: String** The color to start the effect from. See `set_state` for more. default: current bulb color

**period: Double** The time in seconds for one cycles of the effect. default: 1.0

**cycles: Double** The number of times to repeat the effect. default: 1.0

**persist: Boolean** If false set the light back to its previous value when effect ends, if true leave the last effect color. default: false

**power\_on: Boolean** If true, turn the bulb on if it is not already on. default: true

**peak: String** Defines where in a period the target color is at its maximum. Minimum 0.0, maximum 1.0. default: 0.5

**cycle\_lights** (*states*, *defaults*, *direction='forward'*, *selector='all'*)

Cycle through list of effects.

Provide array states as a list of dictionaries with `set_state` arguments. See <http://api.developer.lifx.com/docs/cycle>

**selector: String** The selector to limit which lights will run the effect. default: all

**states: required List of Dicts** List of arguments, named as per `set_state`. Must have 2 to 5 entries.

**defaults: Object** Default values to use when not specified in each `states[]` object. Argument names as per `set_state`.

**direction: String** Direction in which to cycle through the list. Can be forward or backward default: forward

**list\_lights** (*selector='all'*)

Given a selector (defaults to all), return a list of lights. Without a selector provided, return list of all lights.

**list\_scenes** ()

Return a list of scenes. See <http://api.developer.lifx.com/docs/list-scenes>

**pulse\_lights** (*color, selector='all', from\_color=None, period=1.0, cycles=1.0, persist=False, power\_on=True*)

Perform pulse effect on lights.

**selector: String** The selector to limit which lights will run the effect. default: all

**color: required String** Color attributes to use during effect. See `set_state` for more.

**from\_color: String** The color to start the effect from. See `set_state` for more. default: current bulb color

**period: Double** The time in seconds for one cycles of the effect. default: 1.0

**cycles: Double** The number of times to repeat the effect. default: 1.0

**persist: Boolean** If false set the light back to its previous value when effect ends, if true leave the last effect color. default: false

**power\_on: Boolean** If true, turn the bulb on if it is not already on. default: true

**set\_state** (*selector='all', power=None, color=None, brightness=None, duration=None*)

Given a selector (defaults to all), set the state of a light. Selector can be based on id, scene\_id, group\_id, label, etc. Returns list of lightbulb statuses if successful. See <http://api.developer.lifx.com/v1/docs/selectors>

**selector: required String** The selector to limit which lights will run the effect.

**power: String** e.g “on” or “off”

**color: String** e.g #ff0000 or “red” Color to set selected bulbs. Hex color code, color name, saturation percentage, hue, RGB, etc. See <http://api.developer.lifx.com/v1/docs/colors>

**brightness: Double** e.g 0.5 Set brightness level from 0 to 1

**duration: Double** e.g 10 Setting transition time, in seconds, from 0.0 to 3155760000.0 (100 years).

**state\_delta** (*selector='all', power=None, duration=1.0, infrared=None, hue=None, saturation=None, brightness=None, kelvin=None*)

Given a state delta, apply the modifications to lights’ state over a given period of time.

**selector: required String** The selector to limit which lights are controlled.

**power: String** The power state you want to set on the selector. on or off

**duration: Double** How long in seconds you want the power action to take. Range: 0.0 – 3155760000.0 (100 years)

**infrared: Double** The maximum brightness of the infrared channel.

**hue: Double** Rotate the hue by this angle in degrees.

**saturation: Double** Change the saturation by this additive amount; the resulting saturation is clipped to [0, 1].

**brightness: Double** Change the brightness by this additive amount; the resulting brightness is clipped to [0, 1].

**kelvin: Double** Change the kelvin by this additive amount; the resulting kelvin is clipped to [2500, 9000].

**toggle\_power** (*selector='all', duration=1.0*)

Given a selector and transition duration, toggle lights (on/off)

- `genindex`

- [modindex](#)
- [search](#)





**p**

`pifx`, [1](#)



## A

`activate_scene()` (*pifx.PIFX method*), 1

## B

`breathe_lights()` (*pifx.PIFX method*), 1

## C

`cycle_lights()` (*pifx.PIFX method*), 1

## L

`list_lights()` (*pifx.PIFX method*), 1

`list_scenes()` (*pifx.PIFX method*), 2

## P

`PIFX` (*class in pifx*), 1

`pifx` (*module*), 1

`pulse_lights()` (*pifx.PIFX method*), 2

## S

`set_state()` (*pifx.PIFX method*), 2

`state_delta()` (*pifx.PIFX method*), 2

## T

`toggle_power()` (*pifx.PIFX method*), 2