
Python Hue Client Documentation

Release 1.0.0

Adam Charnock

March 23, 2016

1 Quick Start	1
2 API Reference	3
3 Installation	5
4 Contributing	7

Quick Start

This library is modelled roughly on concepts borrowed from Django's ORM. There are some [examples](#) available in GitHub, but let's dive in with an example that list all the available lights:

```
from pprint import pprint
from hueclient.api import hue_api
from hueclient.models.light import Light

if __name__ == '__main__':
    hue_api.authenticate_interactive(app_name='List Lights Example')

    for light in Light.objects.all():
        print(
            "Light {id} is named '{name}' and is {onoff} (brightness: {brightness})".format(
                id=light.id,
                name=light.name,
                onoff='on' if light.state.on else 'off',
                brightness=light.state.brightness,
            )
        )
```

Here is an example which blinks a specific light:

```
from time import sleep
from hueclient.api import hue_api
from hueclient.models.light import Light

# examples/blink_light.py
if __name__ == '__main__':
    # Make sure we are authenticated with the hue bridge.
    # You will be prompted if no username is found in ~/.python_hue
    hue_api.authenticate_interactive(app_name='Blink Light Example')

    # Get light ID 1
    light = Light.objects.get(id=1)

    # Loop forever
    while True:
        # Flip the on state from on -> off / off -> on
        light.state.on = not light.state.on

        # Save the state back to the bridge
        # (Note: required in order for your changes to take effect)
```

```
light.state.save()

# Pause here for a couple of seconds to create a slow blink
# (Note: It is important to sleep here at least a little to
# avoid overloading the bridge with API requests)
sleep(2)
```

1.1 Digging a Little Deeper

You may have noticed the call to `Light.objects.get()` and `Light.objects.all()` in the above example, but what does ‘objects’ mean?

The `objects` attribute is what we call a ‘manager’. The manager manages access to each type of resource. Every resource will have a manager called `objects`, which is referred to as the default manager.

Some resources have additional managers for your convenience. For example, the `Light` resource also has the `Light.reachable`, `Light.unreachable` and `Light.new` managers. To get a list of all unreachable lights you could use:

```
# Get all unreachable lights
Light.unreachable.all()
```

Now you have the basics, go and checkout the [API Reference](#).

API Reference

Contents:

2.1 Lights API

2.2 Groups API

2.3 Schedules API

2.4 Scenes API

Note: Modifying scenes works differently to most state modifications. See `SceneStateChange` for details.

2.5 Rules API

2.6 Monitoring API

Installation

You can install using either pip (preferred) or easy_install:

```
pip install python-hue-client
```

Or:

```
easy_install python-hue-client
```

Contributing

Contributions are exceptionally welcome! Please submit pull requests and report issues on [GitHub](#).