
hostedpi

Release 0.3.0

Apr 30, 2023

Contents

1	Usage	3
2	Table of Contents	5
3	Contributing	33
	Python Module Index	35
	Index	37

Python interface to the [Mythic Beasts Hosted Pi](#) API, developed by the [piwheels](#) team ([Ben Nuttall](#) and [Dave Jones](#)).

This module provides a Pythonic interface to the API, as well as a command line interface.

The authors of this library are not affiliated with Mythic Beasts, but we use their Pi cloud to power the piwheels project.

Documentation of the API itself can be found at <https://www.mythic-beasts.com/support/api/raspberry-pi>

Note: Note that the library is currently in beta. The API and CLI are not yet stable and may change. Once the library reaches v1.0, it will be considered stable.

CHAPTER 1

Usage

View the information about a Pi from the command line:

```
$ hostedpi show mypi
Name: mypi
Provision status: live
Model: Raspberry Pi 3B
Disk size: 10GB
Power: on
IPv6 address: 2a00:1098:8:5b::1
IPv6 network: 2a00:1098:8:5b00::/56
Initialised keys: yes
SSH keys: 4
IPv4 SSH port: 5091
Location: MER
URLs:
  http://www.mypi.hostedpi.com
  https://www.mypi.hostedpi.com
SSH commands:
  ssh -p 5091 root@ssh.mypi.hostedpi.com # IPv4
  ssh root@[2a00:1098:8:5b::1] # IPv6
```

Provision a new Pi and view its SSH command (using Python):

```
>>> from hostedpi import PiCloud
>>> api_id = '8t29hvcux5g9vud8'
>>> secret = 'QNwsvxZY8SxT3OiLt:Vmz-DlmWQuoZ'
>>> cloud = PiCloud(api_id, secret, ssh_key_path='/home/ben/.ssh/id_rsa.pub')
>>> pi = cloud.create_pi('mypi')
>>> print(pi.ssh_command)
ssh -p 5123 root@ssh.mypi.hostedpi.com
```

See the [Getting started](#) page for information on how to authenticate, and see the [Command line interface](#) page for information on using the command line interface.

2.1 Getting started

This page contains a simple tutorial to help you get started by creating a Mythic Beasts account, create an API key, use the command line interface and the Python module.

2.1.1 Create a Mythic Beasts account

Sign up: <https://www.mythic-beasts.com/user/login>

2.1.2 Create an API key

1. Log in to your Mythic Beasts account: <https://www.mythic-beasts.com/user/login>
2. Open the API Keys page: <https://www.mythic-beasts.com/customer/api-users>
3. Enter a descriptive name for your API key, for your own reference
4. Check the *Raspberry Pi Provisioning* box
5. Click the *Create API key* button
6. Make a note of the API ID and Secret. You'll need them to use this Python module, and you can't retrieve them after this screen is gone.

Note: If you lose your keys, you can simply reset them or create a new API key.

2.1.3 Install the hostedpi module

For a system-wide installation:

```
$ sudo pip3 install hostedpi
```

for a user-level installation:

```
$ pip3 install hostedpi --user
```

or in a virtual environment:

```
$ pip install hostedpi
```

2.1.4 Test your API keys

To test your API connection, try running the following commands in a terminal window, with your API ID and secret:

```
$ HOSTEDPI_ID='YOUR ID' HOSTEDPI_SECRET='YOUR SECRET' hostedpi test
Connected to the Mythic Beasts API
```

This message means your API credentials were found and a successful connection was made.

2.1.5 Start using the Python module

The following Python program will connect to the Mythic Beasts API using your credentials, and print out a list of Pi services in your account:

```
from hostedpi import PiCloud

cloud = PiCloud(api_id='YOUR API ID', secret='YOUR SECRET')

for name in cloud.pis:
    print(name)
```

Note: You can either construct *PiCloud* with your API ID and secret, or set them in environment variables like above.

You can provision a new Pi with the *create_pi()* method:

```
from hostedpi import PiCloud

cloud = PiCloud(api_id='YOUR API ID', secret='YOUR SECRET')

pi = cloud.create_pi('mypi3')
```

The default values are for a Pi 3 with a 10GB disk, but you can request either a Pi 3 or Pi 4 and specify the disk size (which must be a multiple of 10):

```
from hostedpi import PiCloud

cloud = PiCloud(api_id='YOUR API ID', secret='YOUR SECRET')

pi = cloud.create_pi('mypi4', model=4, disk_size=20)
```

Note: When requesting a Pi 3, you will either get a model 3B or 3B+. It is not possible to request a particular model beyond 3 or 4. The Pi 4 is the 4GB RAM model.

The return value of this method is a *Pi* object which you can use to retrieve information about the service, and to manage it. The repr of a *Pi* object includes the name and model:

```
>>> pi
<Pi model 4 mypi4>
```

For example, you can retrieve the SSH command needed to connect to it:

```
print(pi.ssh_command)
```

which should print something like:

```
ssh -p 5123 root@ssh.mypi4.hostedpi.com
```

Other properties you can read are include:

- *boot_progress*
- *disk_size*
- *ipv4_ssh_command*
- *ipv4_ssh_config*
- *ipv4_ssh_port*
- *ssh_keys*
- *url*

There are also methods such as *reboot()*:

```
pi.reboot()
```

Other methods include:

- *cancel()*
- *ssh_import_id()*

2.1.6 More

- See the *Command line interface* page for details of the possibilities provided by ready-made scripts
- See the API documentation for *Pi* for more information on available properties and methods.
- See the *Recipes* page for more ideas showing what you can do with this module.

2.2 Command line interface

The `hostedpi` command is a multi-purpose utility for common actions interacting with the API.

This program requires API keys to be provided using environment variables `HOSTEDPI_ID` and `HOSTEDPI_SECRET`.

The following commands are available:

2.2.1 hostedpi add-key

Add an SSH key from a public key file to one or more Pis

Synopsis

```
hostedpi add-key [-h] ssh_key_path [names [names ...]]
```

Description

ssh_key_path

The path to an SSH public key file to add to the Pi

names [names ...]

The name of the Pis to add keys to

-h, --help

Show this help message and exit

Usage

Add your SSH key to one Pi:

```
$ hostedpi add-key ~/.ssh/id_rsa.pub mypi
1 key added to mypi
```

Add your SSH key to multiple Pis:

```
$ hostedpi add-key ~/.ssh/id_rsa.pub mypi mypi2 pypi3
0 keys added to mypi
1 key added to mypi2
1 key added to mypi3
```

Note: Keys are counted before and after addition, and de-duplicated, so if the key is already found on the Pi, it will show as not having been added, as above.

2.2.2 hostedpi cancel

Cancel one or more Pis in the account

Synopsis

```
hostedpi cancel [-h] [-y] names [names ...]
```

Description

names

The name of the Pis to cancel

-h, --help

Show this help message and exit

-y, --yes

Proceed without confirmation

Usage

Cancel a Pi:

```
$ hostedpi cancel mypi
Cancelling 1 Pi. Proceed? [Y/n]
mypi cancelled
```

Note: You can cancel by entering n or interrupting with Ctrl + C.

Note: Unlike other commands, there is no implicit targeting of all Pis. Pis must be listed explicitly to be cancelled.

Cancel multiple Pis:

```
$ hostedpi cancel mypi mypi2
Cancelling 2 Pis. Proceed? [Y/n]
mypi cancelled
mypi2 cancelled
```

Cancel a Pi without the confirmation step:

```
$ hostedpi cancel mypi -y
mypi cancelled
```

Warning: Be careful!

2.2.3 hostedpi copy-keys

Copy all SSH keys from one Pi to one or more others

Synopsis

```
hostedpi copy-keys [-h] name_src [names_dest [names_dest ...]]
```

Description

name_src

The name of the Pi to copy keys from

names_dest

The names of the Pis to copy keys to

-h, --help

Show this help message and exit

Usage

Copy the keys from one Pi to another:

```
$ hostedpi copy-keys mypi mypi2
2 keys added to mypi2
```

Copy the keys from one Pi to several others:

```
$ hostedpi copy-keys mypi mypi2 mypi3 mypi4
0 keys added to mypi2
2 keys added to mypi3
1 keys added to mypi4
```

Note: Keys are counted before and after addition, and de-duplicated, so if a key is already found on the Pi, it will show as not having been added, as above.

2.2.4 hostedpi count-keys

Show the number of SSH keys currently on one or more Pis

Synopsis

```
hostedpi count-keys [-h] [names [names ...]]
```

Description

names

The names of the Pis to count keys for

-h, --help

Show this help message and exit

Usage

Show the number of keys on a Pi:

```
$ hostedpi count-keys mypi
mypi: 4 keys
```

Show the number of keys on multiple Pis:

```
$ hostedpi count-keys mypi mypi2
mypi: 4 keys
mypi2: 2 keys
```

Note: If no names of Pis are given, the key count will be shown for all Pis in the account

2.2.5 hostedpi create

Provision a new Pi in the account

Synopsis

```
hostedpi create [-h] [--model [model]] [--disk [disk]] [--image [image]]
                [--ssh-key-path [ssh_key_path]]
                name
```

Description

name

The name of the new Pi to provision

-h, --help

Show this help message and exit

--model [model]

The model of the new Pi to provision (3 or 4)

--disk [disk]

The disk size in GB

--image [image]

The operating system image to use

--ssh-key-path [ssh key path]

The path to an SSH public key file to add to the Pi

Usage

Provision a new Pi using the default settings:

```
$ hostedpi create mypi
Pi mypi provisioned successfully
```

Provision a new Pi using custom settings:

```
$ hostedpi create mypi4 --model 4 --disk 60 --image ubuntu20.04.arm64 --ssh-key-path ~
↪/.ssh/id_rsa.pub
Pi mypi4 provisioned successfully
```

Note: Use the *hostedpi images* command to retrieve the available operating system images for each Pi model.

Note: More information about the Pi will be available with the command *hostedpi show* once it's finished provisioning.

2.2.6 hostedpi images

Retrieve the list of operating system images available for the given Pi model

Synopsis

```
hostedpi images [-h] [model]
```

Description

model

The Pi model number (3 or 4) to get operating systems for

-h, --help

Show this help message and exit

Usage

List the available operating system images for Pi 3 and Pi 4:

```
$ hostedpi images
Images for Pi 3:
Ubuntu 20.04 (Focal Fossa) (32 bit)           : rpi-focal-armhf
Ubuntu 18.04 (Bionic Beaver) (64 bit)         : rpi-bionic-arm64
Ubuntu 20.04 (Focal Fossa) (64 bit)           : rpi-focal-arm64
Ubuntu 22.04 (Jammy Jellyfish) (64 bit)        : rpi-jammy-arm64
Raspberry Pi OS Bullseye (32 bit)              : rpi-bullseye-armhf
Raspberry Pi OS Buster (32 bit)                : rpi-buster-armhf
Raspberry Pi OS Buster (64 bit)                : rpi-buster-arm64
Raspberry Pi OS Bullseye (64 bit)              : rpi-bullseye-arm64
Raspberry Pi OS Bullseye Desktop (64 bit, 1920x1080) : rpi-bullseye-arm64-vnc.2022-03-
↪25T17:23:56+00:00

Images for Pi 4:
Ubuntu 20.04 (Focal Fossa) (32 bit)           : rpi-focal-armhf
Raspberry Pi OS Bullseye Desktop (64 bit, 1920x1080) : rpi-bullseye-arm64-vnc.2022-03-
↪25T17:23:56+00:00
Raspberry Pi OS Bullseye (32 bit)              : rpi-bullseye-armhf
Raspberry Pi OS Buster (32 bit)                : rpi-buster-armhf
Ubuntu 20.04 (Focal Fossa) (64 bit)           : rpi-focal-arm64
Ubuntu 18.04 (Bionic Beaver) (64 bit)         : rpi-bionic-arm64
Raspberry Pi OS Bullseye (64 bit)              : rpi-bullseye-arm64
Raspberry Pi OS Buster (64 bit)                : rpi-buster-arm64
Ubuntu 22.04 (Jammy Jellyfish) (64 bit)        : rpi-jammy-arm64
```


List the available operating system images for Pi 3:

```
$ hostedpi images 3
Images for Pi 3:
Ubuntu 20.04 (Focal Fossa) (32 bit)           : rpi-focal-armhf
Ubuntu 18.04 (Bionic Beaver) (64 bit)        : rpi-bionic-arm64
Ubuntu 20.04 (Focal Fossa) (64 bit)          : rpi-focal-arm64
Ubuntu 22.04 (Jammy Jellyfish) (64 bit)      : rpi-jammy-arm64
Raspberry Pi OS Bullseye (32 bit)            : rpi-bullseye-armhf
Raspberry Pi OS Buster (32 bit)              : rpi-buster-armhf
Raspberry Pi OS Buster (64 bit)              : rpi-buster-arm64
Raspberry Pi OS Bullseye (64 bit)            : rpi-bullseye-arm64
Raspberry Pi OS Bullseye Desktop (64 bit, 1920x1080) : rpi-bullseye-arm64-vnc.2022-03-
↪25T17:23:56+00:00
```

List the available operating system images for Pi 4:

```
$ hostedpi images 4
Images for Pi 4:
Ubuntu 20.04 (Focal Fossa) (32 bit)           : rpi-focal-armhf
Raspberry Pi OS Bullseye Desktop (64 bit, 1920x1080) : rpi-bullseye-arm64-vnc.2022-03-
↪25T17:23:56+00:00
Raspberry Pi OS Bullseye (32 bit)            : rpi-bullseye-armhf
Raspberry Pi OS Buster (32 bit)              : rpi-buster-armhf
Ubuntu 20.04 (Focal Fossa) (64 bit)          : rpi-focal-arm64
Ubuntu 18.04 (Bionic Beaver) (64 bit)        : rpi-bionic-arm64
Raspberry Pi OS Bullseye (64 bit)            : rpi-bullseye-arm64
Raspberry Pi OS Buster (64 bit)              : rpi-buster-arm64
Ubuntu 22.04 (Jammy Jellyfish) (64 bit)      : rpi-jammy-arm64
```

Note: The right hand column represents the image label which can be used when provisioning a new Pi with *hostedpi create* and *create_pi()*.

2.2.7 hostedpi keys

Show the SSH keys currently on a Pi

Synopsis

```
hostedpi hostedpi keys [-h] name
```

Description

name

The name of the Pi to show keys for

-h, --help

Show this help message and exit

Usage

Show the SSH keys currently on a Pi:

```
$ hostedpi keys mypi
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDSkt3A1j89RT/540ghIMHXIVwNlAEM3WtmqVG7YN/
↪wYwtsJ8iCszg4/lXQsfLFxYmEVe8L9atgtMGCi5QdYP14X/
↪c+5YxFfm88Yjfx+2xEgUdOr864eaI22yaNMQ0AlyilmK+PcSyxKP4dzkf6B5Nsw8lhfb5n9F5md6GHLLjOGuBbHYlesKJKnt2ch
↪sIWzD6i2S9szDy6aZ alice@gonzo
```

Save the output into a file:

```
$ hostedpi keys mypi > keys.txt
```

2.2.8 hostedpi list

List all Pis in the account

Synopsis

```
hostedpi list [-h]
```

Description

-h, --help

Show this help message and exit

Usage

List all Pis in the account:

```
$ hostedpi list
mypi
mypi2
mypi3
mypi4
```

2.2.9 hostedpi off

Power off one or more Pis in the account

Synopsis

```
hostedpi off [-h] [names [names ...]]
```

Description

names [names ...]
The name of the Pi to power off

-h, --help
Show this help message and exit

Usage

Power off a Pi:

```
$ hostedpi off mypi
mypi powered off
```

Power off multiple Pis:

```
$ hostedpi off mypi mypi2
mypi powered off
mypi2 powered off
```

Note: If no names of Pis are given, all Pis in the account will be powered off

2.2.10 hostedpi on

Power on one or more Pis in the account

Synopsis

```
hostedpi on [-h] [names [names ...]]
```

Description

names [names ...]
The name of the Pi to power on

-h, --help
Show this help message and exit

Usage

Power on a Pi:

```
$ hostedpi on mypi
mypi powered on
```

Power on multiple Pis:

```
$ hostedpi on mypi mypi2
mypi powered on
mypi2 powered on
```

Note: If no names of Pis are given, all Pis in the account will be powered on

2.2.11 hostedpi power

Get the power status for one or more Pis

Synopsis

```
hostedpi power [-h] [names [names ...]]
```

Description

names [names ...]

The names of the Pis to get the power status for

-h, --help

Show this help message and exit

Usage

Get the power status for a Pi:

```
$ hostedpi power mypi
mypi: powered off
```

Show the number of keys on multiple Pis:

```
$ hostedpi power mypi mypi2 mypi3
mypi: powered off
mypi2: powered on
mypi3: powered on
```

Note: If no names of Pis are given, the power status will be shown for all Pis in the account

2.2.12 hostedpi reboot

Reboot one or more Pis in the account

Synopsis

```
hostedpi reboot [-h] [names [names ...]]
```

Description

names [names ...]
The names of the Pis to reboot

-h, --help
Show this help message and exit

Usage

Reboot a Pi:

```
$ hostedpi reboot mypi
mypi rebooted
```

Reboot multiple Pis:

```
$ hostedpi reboot mypi mypi2
mypi rebooted
mypi2 rebooted
```

Note: If no names of Pis are given, all Pis in the account will be rebooted

2.2.13 hostedpi remove-keys

Remove all SSH keys from one or more Pis

Synopsis

```
hostedpi remove-keys [-h] names [names ...]
```

Description

names [names ...]
The name of the Pis to remove keys from

-h, --help
Show this help message and exit

Usage

Remove all SSH keys from a Pi:

```
$ hostedpi remove-keys mypi
2 keys removed from mypi
```

Remove all SSH keys from multiple Pis:

```
$ hostedpi remove-keys mypi mypi2
2 keys removed from mypi
0 keys removed from mypi2
```

Note: Unlike other commands, there is no implicit targeting of all Pis. Pis must be listed explicitly to have keys removed.

2.2.14 hostedpi show

Show the information about one or more Pis in the account

Synopsis

```
hostedpi show [-h] [names [names ...]]
```

Description

names [names ...]

The names of the Pis to show information for

-h, --help

Show this help message and exit

Usage

Show the information about a Pi:

```
$ hostedpi show mypi
Name: mypi
Provision status: live
Model: Raspberry Pi 3B
Disk size: 10GB
Power: off
IPv6 address: 2a00:1098:8:5b::1
IPv6 network: 2a00:1098:8:5b00::/56
Initialised keys: yes
SSH keys: 0
IPv4 SSH port: 5091
Location: MER
URLs:
  http://www.mypi.hostedpi.com
  https://www.mypi.hostedpi.com
SSH commands:
  ssh -p 5091 root@ssh.mypi.hostedpi.com # IPv4
  ssh root@[2a00:1098:8:5b::1] # IPv6
```

Show the number of keys on multiple Pis:

```
$ hostedpi show mypi mypi2
Name: mypi
Provision status: live
Model: Raspberry Pi 3B
Disk size: 10GB
Power: on
IPv6 address: 2a00:1098:8:5b::1
IPv6 network: 2a00:1098:8:5b00::/56
Initialised keys: yes
SSH keys: 0
IPv4 SSH port: 5091
Location: MER
URLs:
  http://www.mypi.hostedpi.com
  https://www.mypi.hostedpi.com
SSH commands:
  ssh -p 5091 root@ssh.mypi.hostedpi.com # IPv4
  ssh root@[2a00:1098:8:5b::1] # IPv6

Name: mypi2
Provision status: live
Model: Raspberry Pi 4B
Disk size: 40GB
Power: on
IPv6 address: 2a00:1098:8:68::1
IPv6 network: 2a00:1098:8:6800::/56
Initialised keys: yes
SSH keys: 0
IPv4 SSH port: 5072
Location: MER
URLs:
  http://www.mypi2.hostedpi.com
  https://www.mypi2.hostedpi.com
SSH commands:
  ssh -p 5072 root@ssh.mypi2.hostedpi.com # IPv4
  ssh root@[2a00:1098:8:68::1] # IPv6
```

Note: If no names of Pis are given, information about all Pis will be shown.

2.2.15 hostedpi ssh-command

Output the SSH command for one or more Pis in the account

Synopsis

```
hostedpi ssh-command [-h] [--ipv6] [names [names ...]]
```

Description

names [names ...]

The names of the Pis to get SSH commands for

-h, --help
Show this help message and exit

--ipv6
Show IPv6 command

Usage

Output the IPv4 SSH command for a Pi:

```
$ hostedpi ssh-command mypi
ssh -p 5091 root@ssh.mypi.hostedpi.com
```

Output the IPv6 SSH command for a Pi:

```
$ hostedpi ssh-command mypi --ipv6
ssh root@[2a00:1098:8:5b::1]
```

Show the number of keys on multiple Pis:

```
$ hostedpi ssh-command mypi mypi2
ssh -p 5091 root@ssh.mypi.hostedpi.com
ssh -p 5091 root@ssh.mypi2.hostedpi.com
```

Note: If no names of Pis are given, the SSH commands will be shown for all Pis in the account

Execute the SSH command directly:

2.2.16 hostedpi ssh-config

Output the SSH config for one or more Pis in the account

Synopsis

```
hostedpi ssh-config [-h] [--ipv6] [names [names ...]]
```

Description

names [names ...]
The name of the Pis to add keys to

-h, --help
Show this help message and exit

--ipv6
Use IPv6 connection details

Usage

Output the IPv4 SSH config for a Pi:

```
$ hostedpi ssh-config mypi
Host mypi
    user root
    port 5224
    hostname ssh.mypi.hostedpi.com
```

Output the IPv6 SSH config for a Pi:

```
$ hostedpi ssh-config mypi --ipv6
Host mypi
    user root
    hostname 2a00:1098:8:5b::1
```

Output the IPv4 SSH config for multiple Pis:

```
$ hostedpi ssh-config mypi mypi2
Host mypi
    user root
    port 5224
    hostname ssh.mypi.hostedpi.com
Host mypi2
    user root
    port 5072
    hostname ssh.mypi2.hostedpi.com
```

Note: If no names of Pis are given, the SSH commands will be shown for all Pis in the account

Save (append) the IPv4 SSH config for all Pis in the account into your SSH config file:

```
$ hostedpi ssh-config >> ~/.ssh/config
```

Note: Read more about the SSH config file: <https://www.ssh.com/ssh/config/>

2.2.17 hostedpi ssh-import-id

Import SSH keys from GitHub or Launchpad and add them to one or more Pis

Synopsis

```
hostedpi ssh-import-id [-h] [--gh [github username]] [--lp [launchpad username]]
                        [names [names ...]]
```

Description

names [names ...]

The names of the Pis to import keys onto

-h, --help

Show this help message and exit

--gh [github username]

The GitHub username to import keys from

--lp [launchpad username]

The Launchpad username to import keys from

Usage

Import keys from GitHub onto a Pi:

```
$ hostedpi ssh-import-id mypi --gh bennuttall
4 keys retrieved from GitHub

4 keys added to mypi
```

Import keys from GitHub onto multiple Pis:

```
$ hostedpi ssh-import-id mypi mypi2 --gh bennuttall
4 keys retrieved from GitHub

0 keys added to mypi
4 keys added to mypi2
```

Note: If no names of Pis are given, the key count will be shown for all Pis in the account

Note: Keys are counted before and after addition, and de-duplicated, so if a key is already found on the Pi, it will show as not having been added, as above.

Import keys from GitHub and Launchpad onto a Pi:

```
$ hostedpi ssh-import-id mypi --gh bennuttall --lp bennuttall
4 keys retrieved from GitHub
1 key retrieved from Launchpad

1 key added to mypi
```

Note: Keys are counted before and after addition, and de-duplicated, so if a key is already found on the Pi, it will show as not having been added, as above.

2.2.18 hostedpi status

Get the provision status of one or more Pis

Synopsis

```
hostedpi status [-h] [names [names ...]]
```

Description

names [names ...]

The names of the Pis to get the provision status for

-h, --help

Show this help message and exit

Usage

Get the provision status of a Pi:

```
$ hostedpi status mypi
mypi: live
```

Get the provision status of multiple Pis:

```
$ hostedpi status mypi mypi2
mypi: live
mypi2: provisioning
```

Note: If no names of Pis are given, the provision status will be shown for all Pis in the account

2.2.19 hostedpi test

Test a connection to the Mythic Beasts API using API ID and secret in environment variables

Synopsis

```
hostedpi test [-h]
```

Description

-h, --help

Show this help message and exit

Usage

Successful test:

```
$ hostedpi test
Connected to the Mythic Beasts API
```

An error will be shown if the connection fails:

```
$ hostedpi test
hostedpi error: Failed to authenticate
```

Note: See the *Getting started* section for details on how to authenticate

2.3 Recipes

This page includes some recipes for using the Python library to make custom programs with the utilities provided, perhaps combined with other libraries. See the *Python library API* page for a full API reference.

Note: You'll need to create an API key to be able to use these recipes. See the *Getting started* page to begin. The following examples assume the API keys are set using environment variables, but they can be provided as arguments to the *PiCloud* class constructor.

2.3.1 Provisioning Pis

Provision a Pi

```
from hostedpi import PiCloud

cloud = PiCloud()

pi = cloud.create_pi('mypi', model=3, disk=10)
```

Provision some Pis

```
from hostedpi import PiCloud

cloud = PiCloud()

pis = [
    cloud.create_pi('mypi{}'.format(n), model=3, disk=10)
    for n in range(10)
]
```

Push button to provision a Pi

```
from hostedpi import PiCloud
from gpiozero import Button, LED

cloud = PiCloud()
btn = Button(2)
led = LED(3)

def make_pi():
```

(continues on next page)

(continued from previous page)

```
name = "helloworld"  
cloud.create_pi(name)  
led.on()  
  
btn.when_pressed = make_pi
```

See a live demo at https://twitter.com/ben_nuttall/status/1300442981779025921

Note: This requires the [gpiozero](#) library.

2.3.2 Retrieving data about Pis

List all Pis

```
from hostedpi import PiCloud  
  
cloud = PiCloud()  
  
for name in cloud.pis:  
    print(name)
```

List all Pis and their IPv6 address

```
from hostedpi import PiCloud  
  
cloud = PiCloud()  
  
for name, pi in cloud.pis.items():  
    print(name, pi.ipv6_address)
```

2.3.3 Rebooting

Reboot all Pis

```
from hostedpi import PiCloud  
  
cloud = PiCloud()  
  
for pi in cloud.pis.values():  
    pi.reboot()
```

2.3.4 Power on/off

Boot all Pis powered off

```
from hostedpi import PiCloud

cloud = PiCloud()

for pi in cloud.pis.values():
    if not pi.power:
        pi.on()
```

2.3.5 SSH

List SSH commands for all Pis

```
from hostedpi import PiCloud

cloud = PiCloud()

for pi in cloud.pis.values():
    print(pi.ipv4_ssh_command)
```

```
from hostedpi import PiCloud

cloud = PiCloud()

for pi in cloud.pis.values():
    print(pi.ipv6_ssh_command)
```

Write SSH config to a file

```
from hostedpi import PiCloud

cloud = PiCloud()

with open('config', 'w') as f:
    f.write(cloud.ssh_config)
```

2.3.6 Web

Retrieve the contents of the homepage

Note: Note that a web server must be installed on the Pi for the URL to resolve in a web browser, and an SSL certificate must be created for the https URL to resolve.

Print out:

```
from hostedpi import PiCloud
import requests

cloud = PiCloud()
pi = cloud.pis['somepi']
```

(continues on next page)

(continued from previous page)

```
r = requests.get(pi.url)
print(r.text)
```

Save to a file:

```
from hostedpi import PiCloud
import requests

cloud = PiCloud()
pi = cloud.pis['somepi']

r = requests.get(pi.url)
with open('pi.html', 'w') as f:
    f.write(r.text)
```

Access a particular web location

Access `data.json` from the web server, and print out the message value:

```
from hostedpi import PiCloud
import requests

cloud = PiCloud()
pi = cloud.pis['somepi']

url = pi.url + '/data.json'
r = requests.get(url)
data = r.json()
print(data['message'])
```

2.4 Python library API

This page describes the methods and properties available in the *PiCloud* and *Pi* classes and are intended as a useful reference to the functionality provided.

The way to use the module is to import the *PiCloud* class and initialise it with your API keys. This will allow you to provision new Pi services with the *create_pi()* method or access existing services from the *pis* property.

Once you have a connected *PiCloud* instance and access to newly or previously created *Pi* instances, the following API documentation should prove useful to show what you can do with the API via the *hostedpi* module.

You can authenticate by passing your API ID and secret as arguments to the *PiCloud* constructor:

```
>>> from hostedpi import PiCloud
>>> cloud = PiCloud('YOUR ID', 'YOUR SECRET')
```

or set them in environment variables instead:

```
$ export HOSTEDPI_ID='YOUR ID'
$ export HOSTEDPI_SECRET='YOUR SECRET'
```

```
>>> from hostedpi import PiCloud
>>> cloud = PiCloud()
```

2.4.1 PiCloud

```
class hostedpi.picloud.PiCloud(api_id: Optional[str] = None, api_secret: Optional[str]
                               = None, *, ssh_keys: Union[Set[str], List[str], None] =
                               None, ssh_key_path: Optional[str] = None, ssh_import_github:
                               Union[Set[str], List[str], None] = None, ssh_import_launchpad:
                               Union[Set[str], List[str], None] = None)
```

A connection to the Mythic Beasts Pi Cloud API for creating and managing cloud Pi services.

Set up API keys at <https://www.mythic-beasts.com/customer/api-users>

Parameters

- **api_id** (*str or None*) – Your Mythic Beasts API ID (alternatively, the environment variable HOSTEDPI_ID can be used)
- **api_secret** (*str or None*) – Your Mythic Beasts API secret (alternatively, the environment variable HOSTEDPI_SECRET can be used)
- **ssh_keys** (*list or set or None*) – A list/set of SSH key strings (keyword-only argument)
- **ssh_key_path** (*str or None*) – The path to your SSH public key (keyword-only argument)
- **ssh_import_github** (*list or set or None*) – A list/set of GitHub usernames to import SSH keys from (keyword-only argument)
- **ssh_import_launchpad** (*list or set or None*) – A list/set of Launchpad usernames to import SSH keys from (keyword-only argument)

Note: If any SSH keys are provided on class initialisation, they will be used when creating Pis but are overridden by any passed to the `create_pi()` method.

All SSH arguments provided will be used in combination.

`__str__()`

String of information about all the Pis in the account

```
create_pi (name: str, *, model: int = 3, disk_size: int = 10, os_image: Optional[str] = None,
           ssh_keys: Union[Set[str], List[str], None] = None, ssh_key_path: Optional[str] = None,
           ssh_import_github: Union[Set[str], List[str], None] = None, ssh_import_launchpad:
           Union[Set[str], List[str], None] = None) → hostedpi.pi.Pi
```

Provision a new cloud Pi with the specified name, model, disk size and SSH keys. Return a new *Pi* instance.

Parameters

- **name** (*str*) – A unique identifier for the server. This will form part of the hostname for the server, and must consist only of alphanumeric characters and hyphens.
- **model** (*int or None*) – The Raspberry Pi model to provision (3 or 4) - defaults to 3 (keyword-only argument)
- **disk_size** (*int or None*) – The amount of disk space (in GB) attached to the Pi - must be a multiple of 10 - defaults to 10 (keyword-only argument)

- **os_image** (*str*) – The name of the operating system image to boot from. Defaults to `None` which falls back to Mythic’s default (Raspbian/PiOS stable). If given, this must be a string which appears as a key in the return value of `get_operating_systems()` for the relevant Pi model.
- **ssh_keys** (*list or set or None*) – A list/set of SSH key strings (keyword-only argument)
- **ssh_key_path** (*str or None*) – The path to your SSH public key (keyword-only argument)
- **ssh_import_github** (*list or set or None*) – A list/set of GitHub usernames to import SSH keys from (keyword-only argument)
- **ssh_import_launchpad** (*list or set or None*) – A list/set of Launchpad usernames to import SSH keys from (keyword-only argument)

Note: If any SSH keys are provided on class initialisation, they will be used here but are overridden by any passed to this method.

Note: When requesting a Pi 3, you will either get a model 3B or 3B+. It is not possible to request a particular model beyond 3 or 4. The Pi 4 is the 4GB RAM model.

get_operating_systems (*, *model: int*) → Dict[str, str]

Return a dict of operating systems supported by the given Pi *model* (3 or 4). Dict keys are identifiers (e.g. “rpi-buster-armhf”) which can be used when provisioning a new Pi with `create_pi()`; dict values are text labels of the OS/distro names (e.g. “Raspberry Pi OS Bullseye (32 bit)”).

Parameters model (*int*) – The Raspberry Pi model (3 or 4) to get operating systems for (keyword-only argument)

ipv4_ssh_config

A string containing the IPv4 SSH config for all Pis within the account. The contents could be added to an SSH config file for easy access to the Pis in the account.

ipv6_ssh_config

A string containing the IPv6 SSH config for all Pis within the account. The contents could be added to an SSH config file for easy access to the Pis in the account.

pis

A dictionary of *Pi* objects keyed by their names

2.4.2 Pi

class hostedpi.pi.**Pi**

The *Pi* class represents a single Raspberry Pi service in the Mythic Beasts Pi cloud. Initialising a *Pi* object does not provision a new Pi, rather initialisation is for internal construction only.

There are two ways to get access to a *Pi* object: retrieval from the *pis* dictionary; and the return value of `create_pi()` method.

With a *Pi* object, you can access data about that particular Pi service, add SSH keys, reboot it, cancel it and more.

Note: The `Pi` class should not be initialised by the user, only internally within the module.

`__str__()`

A multi-line string of the information about the Pi

`cancel()`

Cancel the Pi service

`off()`

Power the Pi off and return immediately

`on(*, wait: bool = False) → Optional[bool]`

Power the Pi on. If `wait` is `False` (the default), return immediately. If `wait` is `True`, wait until the power on request is completed, and return `True` on success, and `False` on failure.

`reboot(*, wait: bool = False)`

Reboot the Pi. If `wait` is `False` (the default), return `None` immediately. If `wait` is `True`, wait until the reboot request is completed, and return `True` on success, and `False` on failure.

Note: Note that if `wait` is `False`, you can poll for the boot status while rebooting by inspecting the properties `is_booting` and `boot_progress`.

`ssh_import_id(*, github: Union[Set[str], List[str], None] = None, launchpad: Union[Set[str], List[str], None] = None) → Set[str]`

Import SSH keys from GitHub or Launchpad, and add them to the Pi. Return the set of keys added.

Parameters

- **`ssh_import_github(list or set or None)`** – A list/set of GitHub usernames to import SSH keys from (keyword-only argument)
- **`ssh_import_launchpad(list or set or None)`** – A list/set of Launchpad usernames to import SSH keys from (keyword-only argument)

`boot_progress`

A string representing the Pi's boot progress. Can be `booted`, `powered off` or a particular stage of the boot process if currently booting.

`disk_size`

The Pi's disk size in GB

`initialised_keys`

A boolean representing whether or not the Pi has been initialised with SSH keys

`ipv4_ssh_command`

The SSH command required to connect to the Pi over IPv4

`ipv4_ssh_config`

A string containing the IPv4 SSH config for the Pi. The contents could be added to an SSH config file for easy access to the Pi.

`ipv4_ssh_port`

The SSH port to use when connecting via the IPv4 proxy

`ipv6_address`

The Pi's IPv6 address as an `IPv6Address` object

`ipv6_network`

The Pi's IPv6 network as an `IPv6Network` object

ipv6_ssh_command

The SSH command required to connect to the Pi over IPv6

ipv6_ssh_config

A string containing the IPv6 SSH config for the Pi. The contents could be added to an SSH config file for easy access to the Pi.

is_booting

A boolean representing whether or not the Pi is currently booting

location

The Pi's physical location (data centre)

model

The Pi's model (3 or 4)

model_full

The Pi's model name (3B, 3B+ or 4B)

name

The name of the Pi service

power

A boolean representing whether or not the Pi is currently powered on

provision_status

A string representing the provision status of the Pi. Can be "provisioning", "initialising" or "live".

ssh_keys

Retrieve the SSH keys on the Pi, or use assignment to update them. Property value is a set of strings. Assigned value should also be a set of strings.

url

The http version of the hostedpi.com URL of the Pi.

Note: Note that a web server must be installed on the Pi for the URL to resolve in a web browser.

url_ssl

The https version of the hostedpi.com URL of the Pi.

Note: Note that a web server must be installed on the Pi for the URL to resolve in a web browser, and an SSL certificate must be created. See <https://letsencrypt.org/>

2.5 Development

This page contains reference material for those interested in developing and contributing to the **hostedpi** module.

The project source code is hosted on GitHub at <https://github.com/piwheels/hostedpi> which also includes the [issue tracker](#).

2.5.1 Setting up for Development

1. Clone the repository and enter the directory:

```
$ git clone https://github.com/piwheels/hostedpi  
$ cd hostedpi
```

2. Create a virtual environment e.g. using `virtualenvwrapper`:

```
$ mkvirtualenv hostedpi
```

3. Install the project for development:

```
$ make develop
```

CHAPTER 3

Contributing

- Source code can be found on GitHub at <https://github.com/piwheels/hostedpi>
- Code and documentation contributions welcome
- The issue tracker can be found at <https://github.com/piwheels/hostedpi/issues>
- For issues with the API itself, please contact Mythic Beasts support <https://www.mythic-beasts.com/support>

h

hostedpi, [27](#)

Symbols

`-disk [disk]`
 hostedpi-create command line option, 11

`-gh [github username]`
 hostedpi-ssh-import-id command line option, 22

`-image [image]`
 hostedpi-create command line option, 11

`-ipv6`
 hostedpi-ssh-command command line option, 20
 hostedpi-ssh-config command line option, 20

`-lp [launchpad username]`
 hostedpi-ssh-import-id command line option, 22

`-model [model]`
 hostedpi-create command line option, 11

`-ssh-key-path [ssh key path]`
 hostedpi-create command line option, 11

`-h, -help`
 hostedpi-add-key command line option, 8
 hostedpi-cancel command line option, 9
 hostedpi-copy-keys command line option, 10
 hostedpi-count-keys command line option, 10
 hostedpi-create command line option, 11
 hostedpi-images command line option, 12
 hostedpi-keys command line option, 13

 hostedpi-list command line option, 14
 hostedpi-off command line option, 15
 hostedpi-on command line option, 15
 hostedpi-power command line option, 16
 hostedpi-reboot command line option, 17
 hostedpi-remove-keys command line option, 17
 hostedpi-show command line option, 18
 hostedpi-ssh-command command line option, 19
 hostedpi-ssh-config command line option, 20
 hostedpi-ssh-import-id command line option, 21
 hostedpi-status command line option, 23
 hostedpi-test command line option, 23

`-y, -yes`
 hostedpi-cancel command line option, 9

`__str__()` (*hostedpi.pi.Pi method*), 30

`__str__()` (*hostedpi.picloud.PiCloud method*), 28

B

`boot_progress` (*hostedpi.pi.Pi attribute*), 30

C

`cancel()` (*hostedpi.pi.Pi method*), 30

`create_pi()` (*hostedpi.picloud.PiCloud method*), 28

D

`disk_size` (*hostedpi.pi.Pi attribute*), 30

G

get_operating_systems()
(hostedpi.picloud.PiCloud method), 29

H

hostedpi (*module*), 27

hostedpi-add-key command line option
 -h, -help, 8
 names [names ...], 8
 ssh_key_path, 8

hostedpi-cancel command line option
 -h, -help, 9
 -y, -yes, 9
 names, 9

hostedpi-copy-keys command line option
 -h, -help, 10
 name_src, 10
 names_dest, 10

hostedpi-count-keys command line option
 -h, -help, 10
 names, 10

hostedpi-create command line option
 -disk [disk], 11
 -image [image], 11
 -model [model], 11
 -ssh-key-path [ssh key path], 11
 -h, -help, 11
 name, 11

hostedpi-images command line option
 -h, -help, 12
 model, 12

hostedpi-keys command line option
 -h, -help, 13
 name, 13

hostedpi-list command line option
 -h, -help, 14

hostedpi-off command line option
 -h, -help, 15
 names [names ...], 15

hostedpi-on command line option
 -h, -help, 15
 names [names ...], 15

hostedpi-power command line option
 -h, -help, 16
 names [names ...], 16

hostedpi-reboot command line option
 -h, -help, 17
 names [names ...], 17

hostedpi-remove-keys command line option
 -h, -help, 17
 names [names ...], 17

hostedpi-show command line option

(host-

-h, -help, 18
 names [names ...], 18
 hostedpi-ssh-command command line option
 -ipv6, 20
 -h, -help, 19
 names [names ...], 19
 hostedpi-ssh-config command line option
 -ipv6, 20
 -h, -help, 20
 names [names ...], 20
 hostedpi-ssh-import-id command line option
 -gh [github username], 22
 -lp [launchpad username], 22
 -h, -help, 21
 names [names ...], 21
 hostedpi-status command line option
 -h, -help, 23
 names [names ...], 23
 hostedpi-test command line option
 -h, -help, 23

I

initialised_keys (*hostedpi.pi.Pi attribute*), 30
 ipv4_ssh_command (*hostedpi.pi.Pi attribute*), 30
 ipv4_ssh_config (*hostedpi.pi.Pi attribute*), 30
 ipv4_ssh_config (*hostedpi.picloud.PiCloud attribute*), 29
 ipv4_ssh_port (*hostedpi.pi.Pi attribute*), 30
 ipv6_address (*hostedpi.pi.Pi attribute*), 30
 ipv6_network (*hostedpi.pi.Pi attribute*), 30
 ipv6_ssh_command (*hostedpi.pi.Pi attribute*), 30
 ipv6_ssh_config (*hostedpi.pi.Pi attribute*), 31
 ipv6_ssh_config (*hostedpi.picloud.PiCloud attribute*), 29
 is_booting (*hostedpi.pi.Pi attribute*), 31

L

location (*hostedpi.pi.Pi attribute*), 31

M

model
 hostedpi-images command line option, 12
 model (*hostedpi.pi.Pi attribute*), 31
 model_full (*hostedpi.pi.Pi attribute*), 31

N

name
 hostedpi-create command line option, 11

hostedpi-keys command line option,
 13
 name (*hostedpi.pi.Pi attribute*), 31
 name_src
 hostedpi-copy-keys command line
 option, 10
 names
 hostedpi-cancel command line
 option, 9
 hostedpi-count-keys command line
 option, 10
 names [names ...]
 hostedpi-add-key command line
 option, 8
 hostedpi-off command line option, 15
 hostedpi-on command line option, 15
 hostedpi-power command line option,
 16
 hostedpi-reboot command line
 option, 17
 hostedpi-remove-keys command line
 option, 17
 hostedpi-show command line option,
 18
 hostedpi-ssh-command command line
 option, 19
 hostedpi-ssh-config command line
 option, 20
 hostedpi-ssh-import-id command
 line option, 21
 hostedpi-status command line
 option, 23
 names_dest
 hostedpi-copy-keys command line
 option, 10

O

off() (*hostedpi.pi.Pi method*), 30
 on() (*hostedpi.pi.Pi method*), 30

P

Pi (*class in hostedpi.pi*), 29
 PiCloud (*class in hostedpi.picloud*), 28
 pis (*hostedpi.picloud.PiCloud attribute*), 29
 power (*hostedpi.pi.Pi attribute*), 31
 provision_status (*hostedpi.pi.Pi attribute*), 31

R

reboot() (*hostedpi.pi.Pi method*), 30

S

ssh_import_id() (*hostedpi.pi.Pi method*), 30
 ssh_key_path

hostedpi-add-key command line
 option, 8
 ssh_keys (*hostedpi.pi.Pi attribute*), 31

U

url (*hostedpi.pi.Pi attribute*), 31
 url_ssl (*hostedpi.pi.Pi attribute*), 31