
Power Panel Documentation

Release 1.0.0

David Cheeseman

Sep 03, 2018

Contents

1 Purpose	1
1.1 Running the Server	1
1.2 Deploying Test Device	2
1.3 Use Case Examples	2
2 Indices and tables	5

Purpose

The Power Panel Server Tool is a simple rest service that allows a user to deploy the power_panel.py to raspberry pi devices and plug them into outlets around their home or business to map out breakers with uncertain or missing labels.

1.1 Running the Server

It is best practice to deploy this server onto a network that is either connected to a separate power panel as the one being tested on, a network that is on a UPS, or to a mobile hotspot that allows inner device communication.

First a configuration file must be created. Create a directory called .powerpanel in your home directory then create a config.yml file in that with the following format:

```
# Delay between a devices last registration (sec) allowed before the device is  
↪ removed from the active testers  
# list.  
device_expiration: 3  
# The time after which the homepage should refresh to display the active devices.  
refresh_delay: 1  
# The server port.  
port: 5000  
# NOTE: device_expiration should be at least 2x the refresh_delay.
```

Then run the following command in a python shell.

```
from powerpanel.power_panel import PowerPanelServer  
server = PowerPanelServer()  
server.run()
```

The server can then be reached at [http://\[Server IP\]:\[Server Port\]](http://[Server IP]:[Server Port])

- genindex
- modindex
- search

1.2 Deploying Test Device

Create a raspbian sd card and add a wpa_supplicant.conf for your wifi network and an empty ssh file to initialize ssh so ssh is enabled on the first boot. Boot up the raspberry pi, find it on your network and ssh into it.

Install Python 3.

```
sudo apt install python3 python3-pip
```

Clone the Power Panel project to a directory of your choosing. Then install the package using the following command:

```
pip3 install -e [git repository directory]
```

Create a configuration file.

```
mkdir ~/.powerpanel
cat >~/.powerpanel/config.yml <<EOL
# Human readable name of the device tester.
name: [Name of Tester Device; ex Kitchen Outlet Tester]
# Url of the server.
endpoint_url: \http://[IP of Server]:[Server Port]
# The frequency at which to register the device.
reg_frequency: 1
# NOTE: Must be smaller than the device_expiration time of the server.
EOL
```

Next create a script to run the tester

```
cat >~/.powerpanel/tester.py <<EOL
from powerpanel.power_tester import PowerTester
power_tester = PowerTester()
power_tester.run()
EOL
```

Add the following line to /etc/rc.local just before exit 0:

```
su - pi -c 'python3 ~/.powerpanel/tester.py'
```

Finally reboot the raspberry pi. After it boots you should see the testers name as specified in the yaml in the list of devices. You can then deploy to multiple devices and plug them around the house or business you want to map the outlets to and begin opening breakers and monitoring the list for devices which have disappeared off the server page, then shutting the breaker to verify the device you expect returns.

- genindex
- modindex
- search

1.3 Use Case Examples

1.3.1 Business Use Case Example

You're a property investor that routinely buys up business fronts and rental properties in a historic district that often has power panels with out of date or inaccurate labels. You have in your inventory several raspbery pi's preconfigured with generic names. When you go to a site to update/verify the power panel labels, you move from room to room

plugging in the raspberry pi's into the outlets and keeping track of the generic label to room mapping in a spreadsheet program. After you've plugged in all the devices, you go to the breaker panel and begin cycling breakers, noting which rooms are tied to which breakers in the excel sheet in a new column. After you've cycled all the breakers you simply remove the column with the generic names of the test devices and sort by the breaker number and you have a printable mapping of breaker numbers to rooms.

1.3.2 Home Use Case Example

You have just purchased a house or a condo and after living there for a while you decide to change out the old outlets for newer ones but find that the breaker panel is mostly unlabeled and the few that are there are inaccurate (like I did when I bought my first condo). You have a lot of raspberry pi's handy so you flash them all with rasbian images and deploy the testing utility to them, giving them names of the different rooms in the house. You then plug the respective raspberry pi's into the outlets in the respective rooms. Finally, you launch the server and navigate to it using your phone and go to the breaker panel. One by one you flip the breaker off, watch for a device to disappear off the list, then flip it back on to see it come back and label the panel.

- genindex
- modindex
- search

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`