

---

# home Documentation

*Release 0.3.4*

**Dougal Matthews**

February 01, 2015



<b>1</b>	<b>Quickstart for Installing home</b>	<b>3</b>
1.1	Assumptions! . . . . .	3
1.2	Setting up the Raspberry Pi . . . . .	3
1.3	Installing Home . . . . .	4
1.4	Running Home . . . . .	4
<b>2</b>	<b>Home (home)</b>	<b>5</b>
2.1	home.__init__ . . . . .	5
2.2	home.__main__ . . . . .	5
2.3	home.config . . . . .	5
2.4	home.exceptions . . . . .	6
2.5	home.util . . . . .	6
<b>3</b>	<b>Collect (home.collect)</b>	<b>7</b>
3.1	home.collect.__init__ . . . . .	7
3.2	home.collect.handlers . . . . .	7
3.3	home.collect.loop . . . . .	7
<b>4</b>	<b>Dashboard (home.dash)</b>	<b>9</b>
4.1	home.dash.__init__ . . . . .	9
4.2	home.dash.api . . . . .	9
4.3	home.dash.models . . . . .	10
4.4	home.dash.web . . . . .	10
<b>5</b>	<b>Time Series (home.ts)</b>	<b>13</b>
5.1	home.ts.__init__ . . . . .	13
5.2	home.ts.graph . . . . .	13
5.3	home.ts.models . . . . .	13
	<b>Python Module Index</b>	<b>17</b>



Home is a Python library for home automation. It aims to be a dashboard and control panel for your house. At the moment Home is a fairly new project and the range of supported devices and features is limited.

Contents:



---

## Quickstart for Installing home

---

Home currently makes a large number of assumptions. These steps should work fine if you are using exactly the same setup. If you don't, you are encouraged to [raise an issue](#) on GitHub where we would be happy to try and help! The changes to work on other Unix systems will be relatively minor if you are using the same or similar devices.

### 1.1 Assumptions!

- You are running [Raspbian](#) <sup>1</sup> on a Raspberry Pi <sup>2</sup>.
- You have an [RFXCOM RFXtrx 433](#).
- You have a device supported by the Python library [rfxcom](#) (see a list of supported devices in the README, devices are being added over time.)

### 1.2 Setting up the Raspberry Pi

So, you have a fresh Raspbian setup on your Raspberry Pi? Great. You should be able to use the [bootstrap script](#) in the Home git repository.

A quick summary of what it roughly does, to see in more detail, use the source:

1. apt-get update and upgrade.
2. Install git, postgres, python-dev, pip, screen, supervisor and vim packages with apt.
3. Upgrade pip with pip :)
4. Install virtualenv, virtualenvwrapper and tox with pip
5. Download, compile and install Python 3.4 under /opt/python3.4
6. Add the virtualenvwrapper source to the bashrc
7. Tell postgres to listen on local host. This is obviously the most dangerous, but as this should only be deployed on a private closed network it should be fine.

---

<sup>1</sup> I am running this version: [http://downloads.raspberrypi.org/raspbian\\_latest](http://downloads.raspberrypi.org/raspbian_latest)

<sup>2</sup> I'm using a model B board, the SD card doesn't need to be that big.

## 1.3 Installing Home

Now we are setup, we need to create an environment for Home. First we want to make a Python 3.4 virtualenv. On the pi, run these commands:

```
mkvirtualenv home -p /opt/python3.4/bin/python3.4
pip install home
```

Now you have the code installed, we need to finish setting up the database. You can do this like so:

```
home db upgrade
home create_user admin
```

You can use any username you want and you will be prompted for a password.

## 1.4 Running Home

The best way to do this is via supervisord. To do that, you need to create a config file called `/etc/supervisor/conf.d/home.conf` with the following contents:

```
[program:home-dashboard]
command = /home/pi/.virtualenvs/home/bin/home dashboard
user = pi
autostart = true
autorestart = true
stdout_logfile = /var/log/supervisor/home-dashboard.log
stderr_logfile = /var/log/supervisor/home-dashboard-error.log

[program:home-collect]
command = /home/pi/.virtualenvs/home/bin/home collect --device /dev/serial/by-id/...
user = pi
autostart = true
autorestart = true
stdout_logfile = /var/log/supervisor/home-collect.log
stderr_logfile = /var/log/supervisor/home-collect-error.log
```

You will need to replace the path to your serial device to match what is on your system.

After that, run:

```
sudo service supervisor restart
ps aux | grep home
```

You should see some output showing two processes running.

---

**Note:** If that doesn't appear to work, try restarting supervisor like this:

```
sudo service supervisor stop && sudo service supervisor start
```

For some reason the restart function wasn't working for me.

---

If that worked, you should be able to head the IP of your address on port 5000 in your browser (`http://IP:5000/`). You will be asked to login and then you will see a dashboard.

API Reference Documentation:



---

## Home (home)

---

### 2.1 home.\_\_init\_\_

A simple module that defines how the flask app is created.

```
home.__init__.create_app (config=None)
```

This needs some tidying up. To avoid circular imports we import everything here but it makes this method a bit more gross.

### 2.2 home.\_\_main\_\_

This module is the entry point for the application and is responsible for defining the custom sub commands for the home command line interface.

```
home.__main__.collect (device)
```

```
home.__main__.config_sample ()
```

```
home.__main__.create_user (username)
```

Create a new user.

```
home.__main__.main ()
```

```
home.__main__.nginx_sample ()
```

```
home.__main__.populate_redis ()
```

```
home.__main__.rfxcom (device)
```

Start the event loop to collect data from the serial device.

```
home.__main__.supervisor_sample ()
```

### 2.3 home.config

The default settings.

```
home.config.LOG_FULL_FILENAME = '/var/build/user_builds/home/checkouts/latest/home/home.log'
```

The default location for the full log.

```
home.config.LOG_WARNING_FILENAME = '/var/build/user_builds/home/checkouts/latest/home/home-warning.log'
```

The default location for the warning log.

`home.config.MEDIA_FOLDER = '/var/build/user_builds/home/checkouts/latest/home/dash/static/media'`  
Location of the uploaded/dynamic files.

`home.config.MIGRATE_DIRECTORY = '/var/build/user_builds/home/checkouts/latest/home/migrations'`  
Location of the Alembic database migrations.

`home.config.PACKET_HANDLERS = {'rfxcom.protocol.Elec': 'home.collect.elec_handler', '*': 'home.collect.logging_handler'}`  
The default packet handlers, the devices that will be recorded by default.

`home.config.SQLALCHEMY_DATABASE_URI = 'postgresql://home:home@localhost:5432/home'`  
The database connection string.

`home.config.STATIC_FOLDER = '/var/build/user_builds/home/checkouts/latest/home/dash/static'`  
Location of the dashboard static assets.

`home.config.TEMPLATE_FOLDER = '/var/build/user_builds/home/checkouts/latest/home/dash/templates'`  
Location of the dashboard templates.

## 2.4 home.exceptions

A set of custom exceptions used within Home for different error conditions.

**exception** `home.exceptions.HandlerConfigError`  
Bases: `builtins.Exception`

## 2.5 home.util

The random file all projects have where they dump stuff they don't know where else to place.

`home.util.dtparse(string)`

`home.util.get_or_create(model, **kwargs)`

**class** `home.util.timer(name, log=None)`  
Bases: `contextlib.ContextDecorator`

---

## Collect (home.collect)

---

### 3.1 home.collect.\_\_init\_\_

A set of bound handlers which make the connection between the names returned by python-rfxcom and the time series they should be inserted into.

### 3.2 home.collect.handlers

Handlers are defined to handle events that Home collects. They will either just record the event, or store it in a particular way or take action based on the event.

```
class home.collect.handlers.BaseHandler
    Bases: builtins.object

    format_packet (pkt)

class home.collect.handlers.LoggingHandler
    Bases: home.collect.handlers.BaseHandler

class home.collect.handlers.RecordingHandler (mapping)
    Bases: home.collect.handlers.BaseHandler
```

```
home.collect.handlers.importer (dotted_path)
```

```
home.collect.handlers.load_handlers (handler_mapping)
```

Given a dictionary mapping which looks like the following, import the objects based on the dotted path and yield the packet type and handler as pairs.

If the special string '\*' is passed, don't process that, pass it on as it is a wildcard.

If a non-string object is given for either packet or handler (key or value) assume these are the objects to use and yield them.

```
:: {
    'rfxcom.protocol.Status': 'home.collect.logging_handler', 'rfxcom.protocol.Elec':
    'home.collect.elec_handler', 'rfxcom.protocol.TempHumidity': 'home.collect.temp_humidity_handler',
    '*': 'home.collect.logging_handler' }
```

### 3.3 home.collect.loop

The basic event loop used by Home for running the python-rfxcom integration.

`home.collect.loop.collect` (*dev\_name=None, callbacks=None*)

---

## Dashboard (home.dash)

---

### 4.1 home.dash.\_\_init\_\_

Nothing to see here!

### 4.2 home.dash.api

The web API that is primarily used by the front end for rendering graphs.

```
class home.dash.api.AreasResource
    Bases: home.dash.api.Resource

    methods = ['GET']

    model
        alias of Area

class home.dash.api.DeviceSeriesResource
    Bases: home.dash.api.Resource

    methods = ['GET']

    model
        alias of DeviceSeries

class home.dash.api.DevicesResource
    Bases: home.dash.api.Resource

    methods = ['GET']

    model
        alias of Device

class home.dash.api.GraphsResource
    Bases: home.dash.api.Resource

    methods = ['GET']

    model
        alias of Graph

class home.dash.api.JSONResource
    Bases: flask.views.MethodView

    jsonify_qs (result_sets, **kwargs)
```

```
class home.dash.api.Resource
    Bases: home.dash.api.JSONResource

    get (resource_id=None, name=None)

    methods = ['GET']

    page_size = 1000

class home.dash.api.SearchResource
    Bases: home.dash.api.JSONResource

    get (resource_id=None, name=None)

    methods = ['GET', 'POST']

    post (query=None)

class home.dash.api.SeriesResource
    Bases: home.dash.api.Resource

    methods = ['GET']

    model
        alias of Series

home.dash.api.kwarg_json_query (f)

home.dash.api.register_api (view, endpoint, url)
```

## 4.3 home.dash.models

The models for the web interface, this defines a User to enable authentication.

```
class home.dash.models.User (username, password)
    Bases: flask_sqlalchemy.Model

    check_password (password)

    get_id ()

    id

    is_active ()

    is_anonymous ()

    is_authenticated ()

    password

    registered_on

    set_password (password)

    username
```

## 4.4 home.dash.web

The web controllers for rendering login, logout and dashboard screens.

```
home.dash.web.dashboard (area_name=None)
```

```
home.dash.web.inject_areas()  
home.dash.web.inject_webcams()  
home.dash.web.login()  
home.dash.web.logout()
```





---

## Time Series (home.ts)

---

### 5.1 home.ts.\_\_init\_\_

Nothing to see here!

### 5.2 home.ts.graph

A set of graph aggregation functions that are used to process the results so we can display them in different ways. For example, rather than showing an increasing total watts we can show how much was used in the hour.

```
home.ts.graph.get_method(agggregator_name)
```

```
home.ts.graph.group_values(values, aggregator_functions)
```

```
home.ts.graph.round_datetime(data_point, round_by)
```

### 5.3 home.ts.models

A set of models for storing data points for a time series and device. These can then also be attributed to an Area, which typically defines a room in the house.

```
class home.ts.models.Area(name)
    Bases: flask_sqlalchemy.Model, home.ts.models.SerialiseMixin
    created_at
    devices
    id
    name

class home.ts.models.DataPoint(device_series, value, created_at=None)
    Bases: flask_sqlalchemy.Model, home.ts.models.SerialiseMixin
    created_at
    device_series
    device_series_id
    id
```

```
    push_to_redis()

    classmethod record(series, device, value, created_at=None)

    value

class home.ts.models.Device(device_type, device_sub_type, device_id, area=None)
    Bases: flask_sqlalchemy.Model, home.ts.models.SerialiseMixin

    area

    area_id

    created_at

    device_id

    device_series

    device_sub_type

    device_type

    classmethod get_or_create(device_type, device_sub_type, device_id)

    id

class home.ts.models.DeviceSeries(device, series)
    Bases: flask_sqlalchemy.Model, home.ts.models.SerialiseMixin

    created_at

    data_points

    device

    device_id

    classmethod get_or_create(**kwargs)

    id

    latest_reading

    series

    series_id

class home.ts.models.Graph(**kwargs)
    Bases: flask_sqlalchemy.Model, home.ts.models.SerialiseMixin

    aggregator

    description

    id

    name

    series

class home.ts.models.SerialiseMixin
    Bases: builtins.object

    as_dict()

class home.ts.models.Series(name)
    Bases: flask_sqlalchemy.Model, home.ts.models.SerialiseMixin

    created_at
```

```
device_series
classmethod get_or_create (**kwargs)
graph
graph_id
id
name
```



## h

- home.\_\_init\_\_, 5
- home.\_\_main\_\_, 5
- home.collect.\_\_init\_\_, 7
- home.collect.handlers, 7
- home.collect.loop, 7
- home.config, 5
- home.dash.\_\_init\_\_, 9
- home.dash.api, 9
- home.dash.models, 10
- home.dash.web, 10
- home.exceptions, 6
- home.ts.\_\_init\_\_, 13
- home.ts.graph, 13
- home.ts.models, 13
- home.util, 6



## A

aggregator (home.ts.models.Graph attribute), 14  
Area (class in home.ts.models), 13  
area (home.ts.models.Device attribute), 14  
area\_id (home.ts.models.Device attribute), 14  
AreasResource (class in home.dash.api), 9  
as\_dict() (home.ts.models.SerialiseMixin method), 14

## B

BaseHandler (class in home.collect.handlers), 7

## C

check\_password() (home.dash.models.User method), 10  
collect() (in module home.\_\_main\_\_), 5  
collect() (in module home.collect.loop), 7  
config\_sample() (in module home.\_\_main\_\_), 5  
create\_app() (in module home.\_\_init\_\_), 5  
create\_user() (in module home.\_\_main\_\_), 5  
created\_at (home.ts.models.Area attribute), 13  
created\_at (home.ts.models.DataPoint attribute), 13  
created\_at (home.ts.models.Device attribute), 14  
created\_at (home.ts.models.DeviceSeries attribute), 14  
created\_at (home.ts.models.Series attribute), 14

## D

dashboard() (in module home.dash.web), 10  
data\_points (home.ts.models.DeviceSeries attribute), 14  
DataPoint (class in home.ts.models), 13  
description (home.ts.models.Graph attribute), 14  
Device (class in home.ts.models), 14  
device (home.ts.models.DeviceSeries attribute), 14  
device\_id (home.ts.models.Device attribute), 14  
device\_id (home.ts.models.DeviceSeries attribute), 14  
device\_series (home.ts.models.DataPoint attribute), 13  
device\_series (home.ts.models.Device attribute), 14  
device\_series (home.ts.models.Series attribute), 14  
device\_series\_id (home.ts.models.DataPoint attribute), 13  
device\_sub\_type (home.ts.models.Device attribute), 14  
device\_type (home.ts.models.Device attribute), 14  
devices (home.ts.models.Area attribute), 13

DeviceSeries (class in home.ts.models), 14  
DeviceSeriesResource (class in home.dash.api), 9  
DevicesResource (class in home.dash.api), 9  
dtparse() (in module home.util), 6

## F

format\_packet() (home.collect.handlers.BaseHandler method), 7

## G

get() (home.dash.api.Resource method), 10  
get() (home.dash.api.SearchResource method), 10  
get\_id() (home.dash.models.User method), 10  
get\_method() (in module home.ts.graph), 13  
get\_or\_create() (home.ts.models.Device class method), 14  
get\_or\_create() (home.ts.models.DeviceSeries class method), 14  
get\_or\_create() (home.ts.models.Series class method), 15  
get\_or\_create() (in module home.util), 6  
Graph (class in home.ts.models), 14  
graph (home.ts.models.Series attribute), 15  
graph\_id (home.ts.models.Series attribute), 15  
GraphsResource (class in home.dash.api), 9  
group\_values() (in module home.ts.graph), 13

## H

HandlerConfigError, 6  
home.\_\_init\_\_ (module), 5  
home.\_\_main\_\_ (module), 5  
home.collect.\_\_init\_\_ (module), 7  
home.collect.handlers (module), 7  
home.collect.loop (module), 7  
home.config (module), 5  
home.dash.\_\_init\_\_ (module), 9  
home.dash.api (module), 9  
home.dash.models (module), 10  
home.dash.web (module), 10  
home.exceptions (module), 6  
home.ts.\_\_init\_\_ (module), 13

home.ts.graph (module), 13  
home.ts.models (module), 13  
home.util (module), 6

## I

id (home.dash.models.User attribute), 10  
id (home.ts.models.Area attribute), 13  
id (home.ts.models.DataPoint attribute), 13  
id (home.ts.models.Device attribute), 14  
id (home.ts.models.DeviceSeries attribute), 14  
id (home.ts.models.Graph attribute), 14  
id (home.ts.models.Series attribute), 15  
importer() (in module home.collect.handlers), 7  
inject\_areas() (in module home.dash.web), 10  
inject\_webcams() (in module home.dash.web), 11  
is\_active() (home.dash.models.User method), 10  
is\_anonymous() (home.dash.models.User method), 10  
is\_authenticated() (home.dash.models.User method), 10

## J

jsonify\_qs() (home.dash.api.JSONResource method), 9  
JSONResource (class in home.dash.api), 9

## K

kwargs\_json\_query() (in module home.dash.api), 10

## L

latest\_reading (home.ts.models.DeviceSeries attribute), 14  
load\_handlers() (in module home.collect.handlers), 7  
LOG\_FULL\_FILENAME (in module home.config), 5  
LOG\_WARNING\_FILENAME (in module home.config), 5  
LoggingHandler (class in home.collect.handlers), 7  
login() (in module home.dash.web), 11  
logout() (in module home.dash.web), 11

## M

main() (in module home.\_\_main\_\_), 5  
MEDIA\_FOLDER (in module home.config), 5  
methods (home.dash.api.AreasResource attribute), 9  
methods (home.dash.api.DeviceSeriesResource attribute), 9  
methods (home.dash.api.DevicesResource attribute), 9  
methods (home.dash.api.GraphsResource attribute), 9  
methods (home.dash.api.Resource attribute), 10  
methods (home.dash.api.SearchResource attribute), 10  
methods (home.dash.api.SeriesResource attribute), 10  
MIGRATE\_DIRECTORY (in module home.config), 6  
model (home.dash.api.AreasResource attribute), 9  
model (home.dash.api.DeviceSeriesResource attribute), 9  
model (home.dash.api.DevicesResource attribute), 9  
model (home.dash.api.GraphsResource attribute), 9

model (home.dash.api.SeriesResource attribute), 10

## N

name (home.ts.models.Area attribute), 13  
name (home.ts.models.Graph attribute), 14  
name (home.ts.models.Series attribute), 15  
nginx\_sample() (in module home.\_\_main\_\_), 5

## P

PACKET\_HANDLERS (in module home.config), 6  
page\_size (home.dash.api.Resource attribute), 10  
password (home.dash.models.User attribute), 10  
populate\_redis() (in module home.\_\_main\_\_), 5  
post() (home.dash.api.SearchResource method), 10  
push\_to\_redis() (home.ts.models.DataPoint method), 13

## R

record() (home.ts.models.DataPoint class method), 14  
RecordingHandler (class in home.collect.handlers), 7  
register\_api() (in module home.dash.api), 10  
registered\_on (home.dash.models.User attribute), 10  
Resource (class in home.dash.api), 9  
rfxcom() (in module home.\_\_main\_\_), 5  
round\_datetime() (in module home.ts.graph), 13

## S

SearchResource (class in home.dash.api), 10  
SerialiseMixin (class in home.ts.models), 14  
Series (class in home.ts.models), 14  
series (home.ts.models.DeviceSeries attribute), 14  
series (home.ts.models.Graph attribute), 14  
series\_id (home.ts.models.DeviceSeries attribute), 14  
SeriesResource (class in home.dash.api), 10  
set\_password() (home.dash.models.User method), 10  
SQLALCHEMY\_DATABASE\_URI (in module home.config), 6  
STATIC\_FOLDER (in module home.config), 6  
supervisor\_sample() (in module home.\_\_main\_\_), 5

## T

TEMPLATE\_FOLDER (in module home.config), 6  
timer (class in home.util), 6

## U

User (class in home.dash.models), 10  
username (home.dash.models.User attribute), 10

## V

value (home.ts.models.DataPoint attribute), 14