
Evohome Client Documentation

Release 0.1.0

Andrew Stock

February 08, 2016

1 Purpose	1
2 Getting started	3
3 Versions	5
3.1 Evohome Client - Version 1	5
3.2 Evohome Client - Version 2	7
4 Indices and tables	13

Purpose

The Evohome client provides a straightforward API to allow access to your Evohome data.

Getting started

You'll need to have signed up for an account at <http://www.mytotalconnect.com> - you'll use the username and password credentials below.

Install the evohome client library:

```
pip install ./evohome-client
```

Versions

Honeywell currently have two version of their API.

Version 1 is simpler, but doesn't have capabilities such as viewing and changing the schedules on the evohome system.

Version 2 was released towards the end of October 2014 and has a lot more information available, including the functionality to change the schedules for individual zones.

Contents:

3.1 Evohome Client - Version 1

3.1.1 Instantiating the client

```
from evohomeclient import EvohomeClient

client = EvohomeClient('username', 'password')
```

Contents:

Temperatures

Get all zone temperatures

```
for device in client.temperatures():
    print device
```

Calling this function returns a dictionary for each device which includes the sensor ID, the name of the sensor, the type of sensor and the current temperature reading

The temperatures are cached so if you want to request updated values, you can force a refresh:

```
for device in client.temperatures(force_refresh=True):
    print device
```

Specifying a zone

Zones can be specified either as a string with the name of zone (case sensitive) or based on the ID of the sensor.

Both of these can be found by running the command above to list the current configuration of the system.

```
zone = 'House'  
  
# or  
  
zone = 31234
```

Setting a zone temperature

```
temperature = 19.0  
client.set_temperature(zone, temperature) # set permanent  
  
from datetime import datetime  
client.set_temperature(zone, temperature, datetime(2014,4,10,22,0,0)) # set temperature until 10 Apr
```

Cancelling the temperature override

```
client.cancel_temp_override(zone)
```

Hot Water

Set hot water on

```
client.set_dhw_on() # Permanent  
  
from datetime import datetime  
client.set_dhw_on(datetime(2014,4,10,22,0,0)) # set on until 10 Apr 2014, 10pm
```

Set hot water off

```
client.set_dhw_off() # Permanent  
  
from datetime import datetime  
client.set_dhw_off(datetime(2014,4,10,22,0,0)) # set off until 10 Apr 2014, 10pm
```

Set hot water to auto (cancel override)

```
client.set_dhw_auto()
```

System modes

Evohome supports a number of different modes and the client API has the ability to switch between them.

To set the system back to the normal or auto status call:

```
client.set_status_normal()
```

To enable one of the other modes use one of the calls below. These calls are also able to take a datetime object to specify when to enable the mode until.

Warning: Only the date part of the datetime object will be used.

```
client.set_status_custom() # Use the custom program
client.set_status_eco() # Reduce all temperatures by 3 degrees
client.set_status_away() # Heating and hot water off
client.set_status_dayoff() # Use weekend profile
client.set_status_heatingoff() # Heating off, hot water on
```

3.2 Evohome Client - Version 2

3.2.1 Instantiating the client

```
from evohomeclient2 import EvohomeClient
client = EvohomeClient('username', 'password')
```

To debug the communications, instantiate the client with the debug flag set:

```
from evohomeclient2 import EvohomeClient
client = EvohomeClient('username', 'password', debug=True)
```

Contents:

Temperatures

Get all zone temperatures

```
for device in client.temperatures():
    print device
```

Calling this function returns a dictionary for each device which includes the sensor ID, the name of the sensor, the type of sensor and the current temperature reading

Set a zone temperature

```
ec = EvohomeClient(username, password)
zone = ec.locations[0]._gateways[0]._control_systems[0].zones['Kitchen']
zone.set_temperature(18.0)
# or to specify an end date/time
from datetime import datetime
```

```
dt = datetime(2015, 11, 1, 18, 0, 0)
zone.set_temperature(18.0, dt)
```

Hot Water

Set hot water on

```
client.set_dhw_on() # Permanent

from datetime import datetime
client.set_dhw_on(datetime(2014, 4, 10, 22, 0, 0)) # set on until 10 Apr 2014, 10pm
```

Set hot water off

```
client.set_dhw_off() # Permanent

from datetime import datetime
client.set_dhw_off(datetime(2014, 4, 10, 22, 0, 0)) # set off until 10 Apr 2014, 10pm
```

Set hot water to auto (cancel override)

```
client.set_dhw_auto()
```

System modes

Evohome supports a number of different modes and the client API has the ability to switch between them.

To set the system back to the normal or auto status call:

```
client.set_status_normal()
```

To enable one of the other modes use one of the calls below. These calls are also able to take a datetime object to specify when to enable the mode until.

Warning: Only the date part of the datetime object will be used.

```
client.set_status_custom() # Use the custom program
client.set_status_eco() # Reduce all temperatures by 3 degrees
client.set_status_away() # Heating and hot water off
client.set_status_dayoff() # Use weekend profile
client.set_status_heatingoff() # Heating off, hot water on
```

Note “DayOfWeek” is 0 for Monday, 1 for Tuesday etc.

```
{
  "DailySchedules": [
    {
```

```

    "DayOfWeek": 0,
    "Switchpoints": [
      {
        "TargetTemperature": 21.0,
        "TimeOfDay": "06:30:00"
      },
      {
        "TargetTemperature": 15.0,
        "TimeOfDay": "08:00:00"
      },
      {
        "TargetTemperature": 19.0,
        "TimeOfDay": "16:30:00"
      },
      {
        "TargetTemperature": 15.0,
        "TimeOfDay": "20:00:00"
      }
    ]
  },
  {
    "DayOfWeek": 1,
    "Switchpoints": [
      {
        "TargetTemperature": 21.0,
        "TimeOfDay": "06:30:00"
      },
      {
        "TargetTemperature": 15.0,
        "TimeOfDay": "08:00:00"
      },
      {
        "TargetTemperature": 19.0,
        "TimeOfDay": "16:30:00"
      },
      {
        "TargetTemperature": 15.0,
        "TimeOfDay": "20:00:00"
      }
    ]
  },
  {
    "DayOfWeek": 2,
    "Switchpoints": [
      {
        "TargetTemperature": 21.0,
        "TimeOfDay": "06:30:00"
      },
      {
        "TargetTemperature": 15.0,
        "TimeOfDay": "08:00:00"
      },
      {
        "TargetTemperature": 19.0,
        "TimeOfDay": "16:30:00"
      },
      {
        "TargetTemperature": 15.0,

```

```

        "TimeOfDay": "20:00:00"
    }
]
},
{
    "DayOfWeek": 3,
    "Switchpoints": [
        {
            "TargetTemperature": 21.0,
            "TimeOfDay": "06:30:00"
        },
        {
            "TargetTemperature": 15.0,
            "TimeOfDay": "08:00:00"
        },
        {
            "TargetTemperature": 19.0,
            "TimeOfDay": "16:30:00"
        },
        {
            "TargetTemperature": 15.0,
            "TimeOfDay": "20:00:00"
        },
        {
            "TargetTemperature": 17.5,
            "TimeOfDay": "23:00:00"
        }
    ]
},
{
    "DayOfWeek": 4,
    "Switchpoints": [
        {
            "TargetTemperature": 21.0,
            "TimeOfDay": "06:30:00"
        },
        {
            "TargetTemperature": 15.0,
            "TimeOfDay": "08:00:00"
        },
        {
            "TargetTemperature": 19.0,
            "TimeOfDay": "16:30:00"
        },
        {
            "TargetTemperature": 15.0,
            "TimeOfDay": "20:00:00"
        }
    ]
},
{
    "DayOfWeek": 5,
    "Switchpoints": [
        {
            "TargetTemperature": 21.0,
            "TimeOfDay": "06:30:00"
        },
        {

```

```

        "TargetTemperature": 15.0,
        "TimeOfDay": "08:00:00"
    },
    {
        "TargetTemperature": 19.0,
        "TimeOfDay": "16:30:00"
    },
    {
        "TargetTemperature": 15.0,
        "TimeOfDay": "20:00:00"
    }
]
},
{
    "DayOfWeek": 6,
    "Switchpoints": [
        {
            "TargetTemperature": 21.0,
            "TimeOfDay": "06:30:00"
        },
        {
            "TargetTemperature": 15.0,
            "TimeOfDay": "08:00:00"
        },
        {
            "TargetTemperature": 19.0,
            "TimeOfDay": "16:30:00"
        },
        {
            "TargetTemperature": 15.0,
            "TimeOfDay": "20:00:00"
        }
    ]
}
]
}

```

Backup and Restore

With thanks to Andrew Blake for adding

Backup your zone schedules

```
client.zone_schedules_backup('filename.json')
```

Restore your zone schedules

```
client.zone_schedules_restore('filename.json')
```

Indices and tables

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