A Universal Platform for Building Energy Management

Brian Lauer
Advisor: Dr. Suruz Miah

Department of Electrical and Computer Engineering
Bradley University
1501 W. Bradley Avenue
Peoria, IL, 61625, USA

Friday, November 22, 2019

Outline

1. Objectives
2. Changes to Plan
3. Details on New Device
4. Implementing Motor Scheduling
5. Plans

Objectives

- Narrow down idea for new device
- Implement scheduling feature for motor in BEMOSS

Changes to Plan

- Abandoned plan to integrate stepper motor (little to no application)
- Decided on new device - Sensibo Sky

Figure: Sensibo Sky, courtesy of PCMag.com

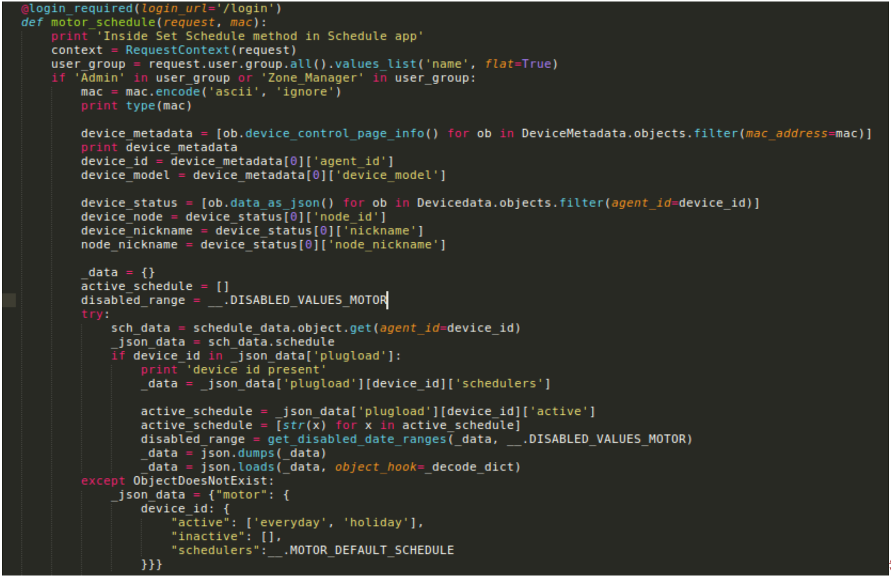
Details on New Device

- Sensibo Sky - Smart air conditioner controller
- Connects to any AC unit through IR remote
- Compatible with IFTTT, Amazon Alexa, Amazon Echo, Google Home
- Capable of measuring temperature, humidity, battery voltage of air conditioner
- Set fan speed, temperature, swing mode, temperature unit (F, C) of air conditioner
- Open API - `https://github.com/kdschlosser/Sensibo_Sky_API.git`

Implementing Motor Scheduling

- Added view function `motor_schedule` in `webapps.schedule.views.py`
- Copied from view function `plugload_schedule` in same file

Implementing Motor Scheduling



```
@login_required(login_url='/login')
def motor_schedule(request, mac):
    print 'Inside Set Schedule method in Schedule app'
    context = RequestContext(request)
    user_group = request.user.group.all().values_list('name', flat=True)
    if 'Admin' in user_group or 'Zone_Manager' in user_group:
        mac = mac.encode('ascii', 'ignore')
        print type(mac)

        device_metadata = [ob.device_control_page_info() for ob in DeviceMetadata.objects.filter(mac_address=mac)]
        print device_metadata
        device_id = device_metadata[0]['agent_id']
        device_model = device_metadata[0]['device_model']

        device_status = [ob.data_as_json() for ob in Devicedata.objects.filter(agent_id=device_id)]
        device_node = device_status[0]['node_id']
        device_nickname = device_status[0]['nickname']
        node_nickname = device_status[0]['node_nickname']

        _data = {}
        active_schedule = []
        disabled_range = __.DISABLED_VALUES_MOTOR
        try:
            sch_data = schedule_data.object.get(agent_id=device_id)
            _json_data = sch_data.schedule
            if device_id in _json_data['plugload']:
                print 'device id present'
                _data = _json_data['plugload'][device_id]['schedulers']

                active_schedule = _json_data['plugload'][device_id]['active']
                active_schedule = [str(x) for x in active_schedule]
                disabled_range = get_disabled_date_ranges(_data, __.DISABLED_VALUES_MOTOR)
                _data = json.dumps(_data)
                _data = json.loads(_data, object_hook=_decode_dict)
        except ObjectDoesNotExist:
            _json_data = {"motor": {
                device_id: {
                    "active": ['everyday', 'holiday'],
                    "inactive": [],
                    "schedulers":__.MOTOR_DEFAULT_SCHEDULE
                }}}
```

Implementing Motor Scheduling

- Exception thrown at line 3 of `motor_schedule`
- ’User’ object has no attribute ’group’
- Code:

  ```
  user_group =
  request.user.group.all().values_list(’name’, flat=True)
  ```

Plans

- Fix issue with scheduling
- Start integrating Sensibo Sky

Any questions?

BRADLEY
University