

Smart Load Solutions
Heating and cooling model API documentation
prototype



2015

Contents

1	Introduction	3
2	Authentication	3
2.1	Device ID creation	3
3	Limits and restrictions	3
4	Temperature sensor data	4
4.1	Description	4
4.2	Message components	4
4.2.1	Required components	4
4.3	Sample messages	4
4.4	Response	4
5	Heating and cooling model optimization requests	4
5.1	Description	4
5.2	Request message components	5
5.2.1	Required components	5
5.2.2	Optional components	5
5.3	Sample messages	5
6	Heating and cooling model optimization responses	6
6.1	Sample message	6

1 Introduction

Heating and cooling model is made for temperature conservation between the desired limits taking into account the electricity price, desired temperatures and limitations. The model consists of heater or cooler which is electrically controlled and two temperature sensors which are used to gather data for optimization.

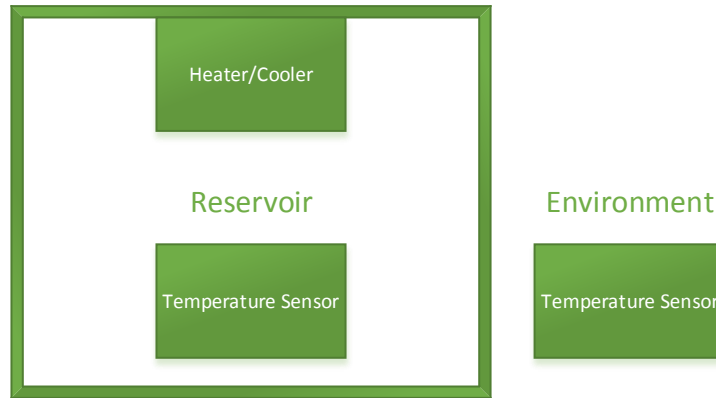


Figure 1: General scheme for heating or cooling

It can be seen from figure 1 that heating or cooling requires 3 components:

- Heater or cooler
- Temperature sensor inside the reservoir
- Temperature sensor outside the reservoir

2 Authentication

2.1 Device ID creation

To be able to send data to our service a key called `deviceID` needs to be created. The UI for deviceID generation is currently under development so the deviceID can be asked directly from SLS.

3 Limits and restrictions

Currently there are no restrictions applied to the service.

4 Temperature sensor data

4.1 Description

Temperature sensor data is gathered to learn the characteristics of the reservoir and heater and cooler. The data should be sent after every 1 to 15 minutes.

4.2 Message components

4.2.1 Required components

- IT – Temperature inside the reservoir °C
- OT – Temperature outside the reservoir in °C
- P – Power of the heater or cooler in MW

4.3 Sample messages

Let us assume that we are heating a house or a room in a house. One temperature sensor is inside the house and the other is outside. Let's say that the temperature sensor inside the room reads that the temperature is 19°C and the sensor outside reads 15°C and the power in the heater is 50kW = 0.05MW. Below is a sample message for sensor data sending.

POST

`https://smartloadsolutions.servicebus.windows.net/sensorinputdata/messages`
HTTP//1.1

Header:

Authorization: SharedAccessSignature sr=smartloadsolutions.servicebus.windows.net
&sig=**deviceID**
&se=1474525966
&skn=SmartMeters
Host: smartloadsolutions.servicebus.windows.net

Body:

`{"ID": "deviceID", "IT":19, "OT":15, "P":0.05}`

The `deviceID` is meant to be replaced by `deviceID` obtained by registering your device.

4.4 Response

If the sending is successful, response with code 201 is sent back.

5 Heating and cooling model optimization requests

5.1 Description

Optimization request is used to get the recommended power plan for the heater or cooler. The optimization requests consists of the requirements and limitations

of the device. The optimization request can be sent once in a day or more frequently i.e. if the limitations change rapidly.

5.2 Request message components

5.2.1 Required components

- **AverageTemperature** – desired temperature for every hour in $^{\circ}\text{C}$;
- **StartTemperature** – temperature at the start of the optimization period in $^{\circ}\text{C}$;
- **TemperaturePenalty** – penalty for the deviation from average temperature in $\frac{\text{€}}{^{\circ}\text{C}}$;
- **MaxHeaterCoolerPower** – maximum power of the heater or cooler in MW;
- **OutsideTemperature**¹ – temperature prediction for the temperature outside of the reservoir in $^{\circ}\text{C}$. This can be sent as an array or single number (this will be the temperature that is assumed to be on every hour).

5.2.2 Optional components

The components below are usually learned by machine learning, but can be overridden by optimization request.

- **heaterCoefficient** – characteristic for the heater or cooler, which describes, how much the room is heated per unit of power $\left[\frac{^{\circ}\text{C}}{\text{MW}}\right]$;
- **heatConductanceCoefficient** – characteristic for the reservoir, which describes, how inside temperature changes according to the temperature gradient on the reservoir edge. This characteristic has no unit.

5.3 Sample messages

Let us use the same example as before with the house heating. Let's say we want to keep an average temperature of 20°C , the current temperature in the room is 19°C , our heater has $100\text{kW} = 0.1\text{MW}$ of power, we have weather prediction data for each hour and let's take penalty for deviation $30\frac{\text{€}}{^{\circ}\text{C}}$. Our sample message in this case would be:

POST

`http://e96537ceb1de467cbd0bec061caa8f51.cloudapp.net/json/reply/OptimizeHeaterCooler`
HTTP//1.1

Header:

Host: `e96537ceb1de467cbd0bec061caa8f51.cloudapp.net`
Content-Type: `application/json`

Body:

¹The temperature prediction is under development so this component will not be required in the future.

```
{
  "averageTemperature":20,"startTemperature":19,"temperaturePenalty":30,
  "maxHeaterCoolerPower":0.1, "OutsideTemperature":
  [10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10],
  "deviceID":"deviceID"
}
```

6 Heating and cooling model optimization responses

After the optimization request has been sent, the power consumption of heater or cooler for each hour is sent back.

- **power** – heater or cooler power consumption for each hour in MW;
- **timeStamps** – timestamps for power (starting time);
- **report** – detailed report of the optimization;

6.1 Sample message

Sample message returned from the service. The report has been cut out for better reading purposes.

```
{
  "power": [0.0025,0.0006,0.0006,0.0026,0,0,0,0.0006,0.0006,0.0006,
  0.0006,0.0006,0.0006,0.0006,0.0076,0,0,0,0,0,0,0,0,
  0,0,0.0006,0.0006,0.0033,0,0,0,0,0.0012,0,0],
  "timeStamps": ["\\/Date(1444302000000-0000)\\/","\\/Date(1444305600000-0000)\\/","
  "\/Date(1444309200000-0000)\\/","\\/Date(1444312800000-0000)\\/","
  "\/Date(1444316400000-0000)\\/","\\/Date(1444320000000-0000)\\/","
  "\/Date(1444323600000-0000)\\/","\\/Date(1444327200000-0000)\\/","
  "\/Date(1444330800000-0000)\\/","\\/Date(1444334400000-0000)\\/","
  "\/Date(1444338000000-0000)\\/","\\/Date(1444341600000-0000)\\/","
  "\/Date(1444345200000-0000)\\/","\\/Date(1444348800000-0000)\\/","
  "\/Date(1444352400000-0000)\\/","\\/Date(1444356000000-0000)\\/","
  "\/Date(1444359600000-0000)\\/","\\/Date(1444363200000-0000)\\/","
  "\/Date(1444366800000-0000)\\/","\\/Date(1444370400000-0000)\\/","
  "\/Date(1444374000000-0000)\\/","\\/Date(1444377600000-0000)\\/","
  "\/Date(1444381200000-0000)\\/","\\/Date(1444384800000-0000)\\/","
  "\/Date(1444388400000-0000)\\/","\\/Date(1444392000000-0000)\\/","
  "\/Date(1444395600000-0000)\\/","\\/Date(1444399200000-0000)\\/","
  "\/Date(1444402800000-0000)\\/","\\/Date(1444406400000-0000)\\/","
  "\/Date(1444410000000-0000)\\/","\\/Date(1444413600000-0000)\\/","
  "\/Date(1444417200000-0000)\\/","\\/Date(1444420800000-0000)\\/"],
  "report": "===Solver Foundation Service Report===..."
}
```