

[◀ Back to Week 4](#)[X Lessons](#)[Prev](#)[Next](#)

Programming Assignment: SenseHAT python API

You have not submitted. You must earn 1/1 points to pass.

Deadline Pass this assignment by May 28, 11:59 PM PDT

Instructions

[My submission](#)[Discussions](#)

Assignment goals

Note: This assignment can be completed without the Sensehat board attached to your Raspberry Pi since we provide an emulator as well.

This assignment builds on the experience from lesson 3. Instead of using NodeRED, we will create a python-based application that uses the IoT Foundation APIs. The application will run on the Raspberry PI to transmit temperature data from the SenseHAT device and react on commands when the threshold is crossed. In order to achieve this from the Raspberry Pi side you will have to implement some glue code in Python.

How to complete the assignment

This assignment provides a skeleton project to get you started. Before you can run the command to install the required packages (detailed in the readme.md file in the zip file) you need to add an additional package to your Raspberry Pi. Please run the following command to install the files needed to be able to build python packages:

```
1 sudo apt-get install -y python-dev
```

Please use the attached zip as a starting point. Download, transfer and unzip the file to a folder on your Raspberry Pi. Follow the instructions in the README.md to install the required packages and then add your code to skeleton.py. The relevant sections in the skeleton are commented as "TODO". Please stick to "display" as the command name and use the same JSON payload as in the previous lesson: { "screen": "off" } and { "screen": "on" }.

iot-coursera-lesson04python-skeleton.zip

This assignment depends on the work you've already done in the last assignment. On Bluemix NodeRED needs to be running with solution from the previous assignment to respond to temperature events with commands. But you have to stop NodeRED on the Pi so that it doesn't interfere with the Python code you are creating. You can use the gateway and device configurations you already have configured in the IBM Watson IoT Platform from the last assignment in this assignment as well. All you need is to replace the following placeholders with the values you get from the IBM Watson IoT Platform Portal.

From the Gateway configuration you have to replace

- yourOrg
- yourGatewayType
- yourGatewayId
- yourAuthToken

From the Device configuration you have to replace

- yourDeviceType
- yourDeviceId

The basic idea is that you read the temperature from the Sensehat device using the python SenseHat API and send it as JSON object to the IBM Watson IoT Platform as device event. Only send temperature data, do not include the humidity data for this assignment, but stick to the same format for the temperature data:

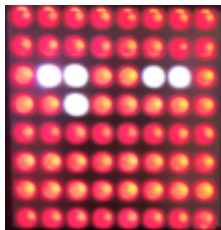
```
1 {"d" : {"temperature" : 29.9}}
```

Note that we are using the Gateway Python API from the IBM Watson IoT Platform, this means your Raspberry Pi basically acts as gateway sending a device event coming from Sensehat in behalf of the sensor. Once you have sent this device Event to the Watson IoT Platform your Node-RED implementation from the last assignment or your Node.js implementation of the next assignment will take care of this and in return sends you back a Device Command. You have to react on this command by completing the implementation of the commandCallbackListener in python.

From there you will turn all LEDs on the Sensehat to either red or green (if you get the command screen on / off) and also encode the temperature as in Assignment 3 of Lecture 3.

For this we have provided a convenience method called "getPixelMap" for you which basically turns the String "green"/"red" and a temperature value into a pixel map which you then can use to control the LEDs on the Sensehat. Here, for your reference the example used in the last Assignment.

Please remember, you not only have to react on the command sent to you via the IBM Watson IoT Platform, you also have to re-read the temperature from Sensehat and encode it on the LEDs.



Note: In case you don't own a SenseHat we have created a SenseHat simulator. The instructions in the `iot-coursera-lesson04python-skeleton.zip` file (README.md) will explain how to use it. Once you've started your application using the simulator you have to wait until you see the following output in order to use the simulator UI:

"New client connected and was given id 1"

Note: The SenseHAT emulator does not contain the full SenseHAT API, so you should only use the following methods:

```
1 set_pixels(pixelMap)
2 get_temperature()
```

The grader uses the emulator to test your code, so even if using a real SenseHAT please limit your use of the API to the above calls or your assignment will fail.

What to submit

Submit `skeleton.py` as `assignment8.py`. Make sure to follow the skeleton code, as the grader uses the defined APIs to test your code. Please also don't use any other imports as already provided in the Skeleton. Please use Python 2.7.

How to submit

When you're ready to submit, you can upload files for each part of the assignment on the "My submission" tab.

