

## **DEVELOP A PYTHON SCRIPT**

To deployment of IOT platform is also initiated by the python interpreter the IOT platform to connect with devices. Few packages need to be installed to work in python interpreter to traverse between simulator and NODE-RED many other services

### **PYTHON CODE FOR NODE-RED AND SIMULATOR**

The below python code communicates between Node-Red Services, Simulator.

#### **CODE:**

```
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random

myConfig = {
    "identity": {
        "orgId": "6fkjbm",
        "typeId": "iotdevice1",
        "deviceId": "qwerty123"
    },
    "auth": {
        "token": "johnyjohnnyespapa"
    }
}

def myCommandCallback(cmd):
```

```

    print("Message received from IBM IoT Platform: %s" %
cmd.data['command'])

    m=cmd.data['command']

    if(m=="Motor-ON"):

        print("*****Motor is Turned
ON*****")

    else:

        print("*****Motor is Turned
OFF*****")

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:

    temp=random.randint(-20,125)
    hum=random.randint(0,100)
    myData={'temperature':temp, 'humidity':hum}
    client.publishEvent(eventId="status", msgFormat="json", data=myData,
qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()

from ibm_watson import TextToSpeechV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator

authenticator =
IAMAuthenticator('M_u6yEvEGJylj_ysbL_pG0ZOKuRCQW1LgXUtv_IcBPC
R')

text_to_speech = TextToSpeechV1(

```

```
    authenticator=authenticator
)

text_to_speech.set_service_url('https://api.au-syd.text-to-
speech.watson.cloud.ibm.com/instances/23724eb6-a096-4a3a-b914-
da0e442c1c5f')

with open('hello_world.wav', 'wb') as audio_file:

    audio_file.write(

        text_to_speech.synthesize(

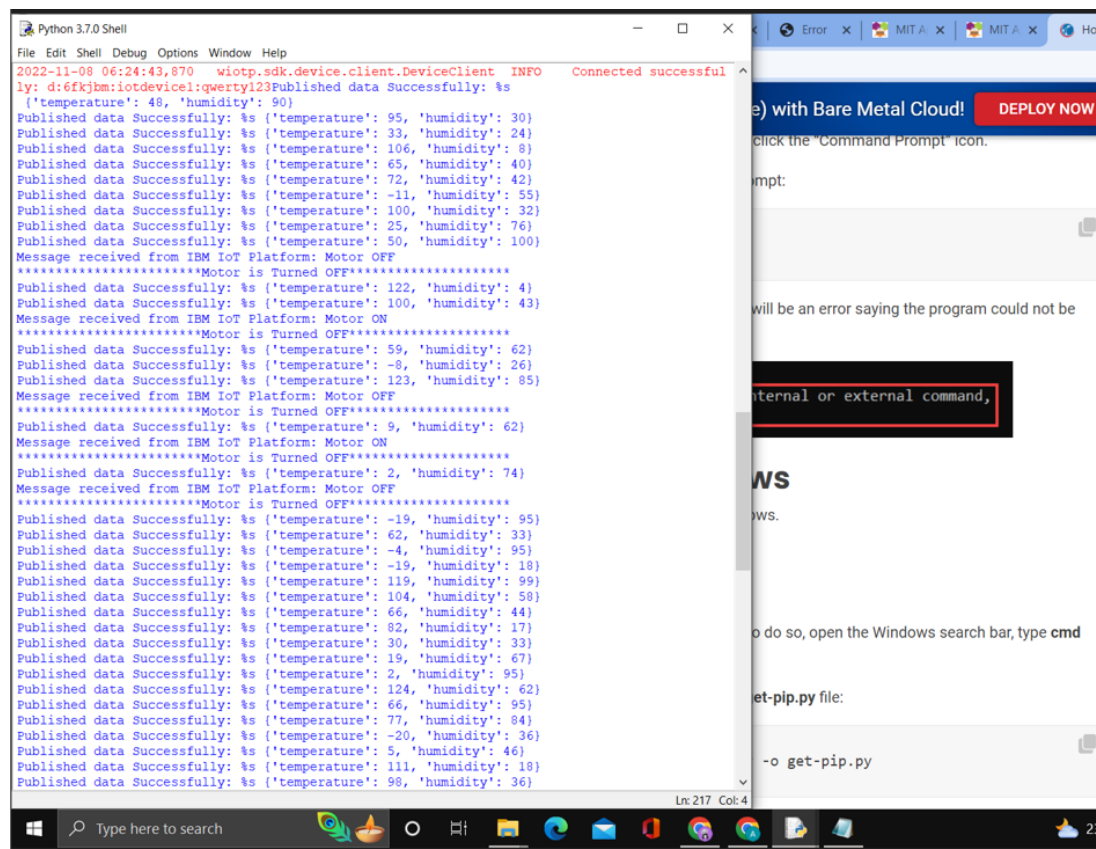
            'Alert',

            voice='en-US_AllisonV3Voice',

            accept='audio/wav'

        ).get_result().content)
```

## OUTPUT:



The screenshot displays a Python 3.7.0 Shell window with the following output:

```
2022-11-08 06:24:43,870 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:6fkjbm:iotdevice1:qwerty123Published data Successfully: %s
({'temperature': 48, 'humidity': 90})
Published data Successfully: %s ('temperature': 95, 'humidity': 30)
Published data Successfully: %s ('temperature': 33, 'humidity': 24)
Published data Successfully: %s ('temperature': 106, 'humidity': 8)
Published data Successfully: %s ('temperature': 65, 'humidity': 40)
Published data Successfully: %s ('temperature': 72, 'humidity': 42)
Published data Successfully: %s ('temperature': -11, 'humidity': 55)
Published data Successfully: %s ('temperature': 100, 'humidity': 32)
Published data Successfully: %s ('temperature': 25, 'humidity': 76)
Published data Successfully: %s ('temperature': 50, 'humidity': 100)
Message received from IBM IoT Platform: Motor OFF
*****Motor is Turned OFF*****
Published data Successfully: %s ('temperature': 122, 'humidity': 4)
Published data Successfully: %s ('temperature': 100, 'humidity': 43)
Message received from IBM IoT Platform: Motor ON
*****Motor is Turned OFF*****
Published data Successfully: %s ('temperature': 59, 'humidity': 62)
Published data Successfully: %s ('temperature': -8, 'humidity': 26)
Published data Successfully: %s ('temperature': 123, 'humidity': 85)
Message received from IBM IoT Platform: Motor OFF
*****Motor is Turned OFF*****
Published data Successfully: %s ('temperature': 9, 'humidity': 62)
Message received from IBM IoT Platform: Motor ON
*****Motor is Turned OFF*****
Published data Successfully: %s ('temperature': 2, 'humidity': 74)
Message received from IBM IoT Platform: Motor OFF
*****Motor is Turned OFF*****
Published data Successfully: %s ('temperature': -19, 'humidity': 95)
Published data Successfully: %s ('temperature': 62, 'humidity': 33)
Published data Successfully: %s ('temperature': -4, 'humidity': 95)
Published data Successfully: %s ('temperature': -19, 'humidity': 18)
Published data Successfully: %s ('temperature': 119, 'humidity': 99)
Published data Successfully: %s ('temperature': 104, 'humidity': 58)
Published data Successfully: %s ('temperature': 66, 'humidity': 44)
Published data Successfully: %s ('temperature': 82, 'humidity': 17)
Published data Successfully: %s ('temperature': 30, 'humidity': 33)
Published data Successfully: %s ('temperature': 19, 'humidity': 67)
Published data Successfully: %s ('temperature': 2, 'humidity': 95)
Published data Successfully: %s ('temperature': 124, 'humidity': 62)
Published data Successfully: %s ('temperature': 66, 'humidity': 95)
Published data Successfully: %s ('temperature': 77, 'humidity': 84)
Published data Successfully: %s ('temperature': -20, 'humidity': 36)
Published data Successfully: %s ('temperature': 5, 'humidity': 46)
Published data Successfully: %s ('temperature': 111, 'humidity': 18)
Published data Successfully: %s ('temperature': 98, 'humidity': 36)
```

On the right, a Windows search bar shows the results for 'get-pip.py'.