

# Real-Time River Water Quality Monitoring and Control Systems

## Develop a python script and upload to ibm cloud

### Code:

```
import random
import time
import sys
import ibmiotf.application
import ibmiotf.device

# Provide your IBM Watson Device Credentials

organization = "uwujz1" # repalce it with organization ID
deviceType = "ibm_iot" # replace it with device type
deviceId = "Python_iot" # repalce with device id
authMethod = "token"
authToken = "1234asdf" # repalce with token

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data)
    if cmd.data['command'] == 'lighton':
        print("LIGHT ON")
    elif cmd.data['command'] == 'lightoff':
        print("LIGHT OFF")

try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,
"auth-method": authMethod,
                    "auth-token": authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)
# .....

except Exception as e:
    print("Caught exception connecting device: %s" % str(e))
    sys.exit()

deviceCli.connect()
```

```

while True:
    pH = random.randint(0,100)
    conductivity = random.randint(0,100)
    T = random.randint(0,100)
    oxygen = random.randint(0,100)
    turbidity = random.randint(0,100)
    # Send Temperature & Humidity to IBM Watson
    data = {"turbidity":turbidity'temp':
T,'ph':pH,'Salinity':conductivity,'oxygen':oxygen}

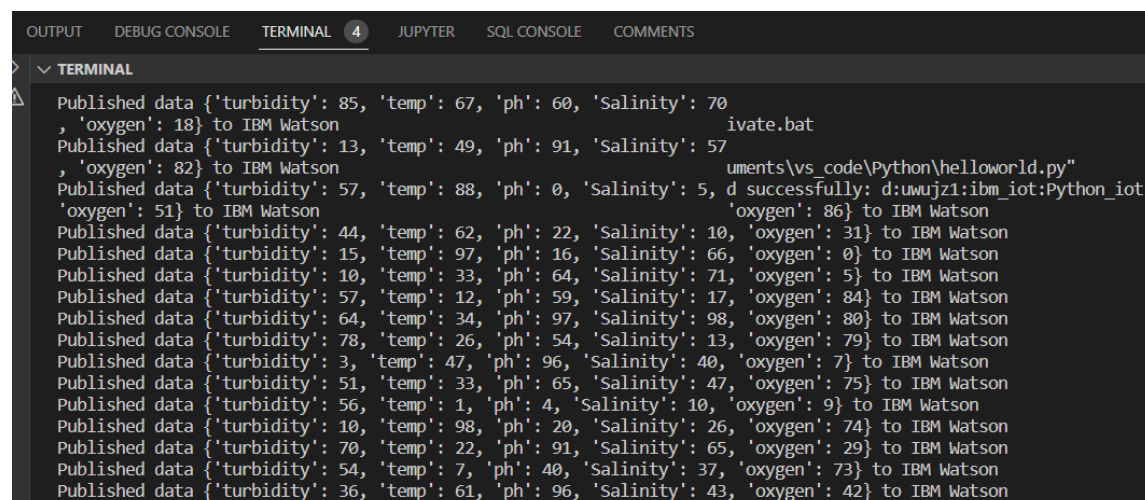
    # print data
    def myOnPublishCallback():
        print("Published data",data, "to IBM Watson")

    success = deviceCli.publishEvent("event", "json", data, 0,
myOnPublishCallback)
    if not success:
        print("Not connected to IoT")
    time.sleep(5)

    deviceCli.commandCallback = myCommandCallback

```

## Output :



```

OUTPUT  DEBUG CONSOLE  TERMINAL  4  JUPYTER  SQL CONSOLE  COMMENTS
>  ▾ TERMINAL
Published data {'turbidity': 85, 'temp': 67, 'ph': 60, 'Salinity': 70
, 'oxygen': 18} to IBM Watson
Published data {'turbidity': 13, 'temp': 49, 'ph': 91, 'Salinity': 57
, 'oxygen': 82} to IBM Watson
Published data {'turbidity': 57, 'temp': 88, 'ph': 0, 'Salinity': 5, d successfully: d:uwujz1:ibm_iot:Python_iot
'oxyge': 51} to IBM Watson
Published data {'turbidity': 44, 'temp': 62, 'ph': 22, 'Salinity': 10, 'oxygen': 31} to IBM Watson
Published data {'turbidity': 15, 'temp': 97, 'ph': 16, 'Salinity': 66, 'oxygen': 0} to IBM Watson
Published data {'turbidity': 10, 'temp': 33, 'ph': 64, 'Salinity': 71, 'oxygen': 5} to IBM Watson
Published data {'turbidity': 57, 'temp': 12, 'ph': 59, 'Salinity': 17, 'oxygen': 84} to IBM Watson
Published data {'turbidity': 64, 'temp': 34, 'ph': 97, 'Salinity': 98, 'oxygen': 80} to IBM Watson
Published data {'turbidity': 78, 'temp': 26, 'ph': 54, 'Salinity': 13, 'oxygen': 79} to IBM Watson
Published data {'turbidity': 3, 'temp': 47, 'ph': 96, 'Salinity': 40, 'oxygen': 7} to IBM Watson
Published data {'turbidity': 51, 'temp': 33, 'ph': 65, 'Salinity': 47, 'oxygen': 75} to IBM Watson
Published data {'turbidity': 56, 'temp': 1, 'ph': 4, 'Salinity': 10, 'oxygen': 9} to IBM Watson
Published data {'turbidity': 10, 'temp': 98, 'ph': 20, 'Salinity': 26, 'oxygen': 74} to IBM Watson
Published data {'turbidity': 70, 'temp': 22, 'ph': 91, 'Salinity': 65, 'oxygen': 29} to IBM Watson
Published data {'turbidity': 54, 'temp': 7, 'ph': 40, 'Salinity': 37, 'oxygen': 73} to IBM Watson
Published data {'turbidity': 36, 'temp': 61, 'ph': 96, 'Salinity': 43, 'oxygen': 42} to IBM Watson

```

## Ibm cloud output:

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains various icons for navigation. The main content area shows a list of devices. The 'Python\_iot' device is selected, and its details are shown in a modal window. The 'Recent Events' tab is active, displaying a table of events.

Event	Value	Format	Last Received
event	{"turbidity":27,"temp":14,"ph":89,"Salinity":33,"..."}	json	a few seconds ago
event	{"turbidity":85,"temp":33,"ph":43,"Salinity":54,"..."}	json	a few seconds ago
event	{"turbidity":89,"temp":40,"ph":21,"Salinity":29,"..."}	json	a few seconds ago
event	{"turbidity":7,"temp":7,"ph":97,"Salinity":27,"oxy..."}	json	a few seconds ago
event	{"temperature":10,"ph":58,"conductivity":99,"ox..."}	json	10 minutes ago