

SPRINT 4

Date	17 November 2022
Team ID	PNT2022TMID32705
Project Name	Real time River Water Quality Monitoring And Control System

STEP 1 :

Python code to connect IBM WATSON

```
import wiotp.sdk.device
```

```
import time
```

```
import random
```

```
import requests
```

```
import os
```

```
from twilio.rest import Client
```

```
myconfig = {
```

```
    "identity": {
```

```
        "orgId": "pyvd3r",
```

```
        "typeId": "waterquality",
```

```
        "deviceId": "1357911"
```

```
    },
```

```
    "auth": {
```

```
        "token": "Aarthi0908"
```

```
    }
```

```
}
```

```
def mycommandCallback(cmd) :
```

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

```
    m=cmd.data[ 'command' ]
```

```
    client = wiotp.sdk.device.DeviceClient(config=myconfig, logHandlers=None)
```

```
    client.connect_apps()
```

```
    count=0
```

```
account_sid = "AC2f35bc97048c9abb2368fc243b07fcbc"
```

```
auth_token = "be94bf37c71bfbc86e8f80542840fe7"
```

```
client = Client(account_sid, auth_token)
```

```
while True:
```

```

temp=random.randint(-20,125)

hum=random.randint(0,100)

pH=random.randint(1,14)

if(temp>90 and pH>8) :

    myData={'temperature':str(temp)+chr(176)+"C", 'humidity':str(hum)+" %",'pH
level':str(pH)+" %", 'condition':"Turn On Water Quality System" }

    message = client.messages \

        .create(

            body='Temperature:'+str(temp)+chr(176)+"C"+"\nHumidity:'+str(hum)+" %"+"\npH
level:'+str(pH)+" %"+"\nCondition:The Water quality Is Not Good",

            from_='+18176705441',

            to='+918610214540'

        )

    print(message.sid)

    print("Turn on Water Qualiy System")

elif(temp<15 and pH>8) :

    myData={'temperature':str(temp)+chr(176)+"C", 'humidity':str(hum)+" %",'pH
level':str(pH)+" %", 'condition':"Turn On Water Quality System" }

    message = client.messages \

        .create(

            body='Temperature:'+str(temp)+chr(176)+"C"+"\nHumidity:'+str(hum)+" %"+"\npH
level:'+str(pH)+" %"+"\nCondition:The Water quality Is Not Good",

            from_='+18176705441',

            to='+918610214540'

        )

    print(message.sid)

    print("Turn on Water Qualiy System")

else :

    myData={'temperature':str(temp)+chr(176)+"C", 'humidity':str(hum)+" %",'pH
level':str(pH)+" %", 'condition':"SAFE" }

    message = client.messages \

        .create(

            body='Temperature:'+str(temp)+chr(176)+"C"+"\nHumidity:'+str(hum)+" %"+"\npH
level:'+str(pH)+" %"+"\nCondition:SAFE DRINK",

            from_='+18176705441',

            to='+918610214540'

        )

```

```
print(message.sid)
```

```
print("SAFE DRINK")
```

```
client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
```

```
print("Published data Successfully: %s", myData)
```

```
client.commandCallback = myCommandCallback
```

```
time.sleep(20)
```

```
client.disconnect()
```

STEP 2 :

Information about the water quality sent from Twilio Account to Higher Authority via SMS

The screenshot displays the Twilio Console's API Explorer interface. The left sidebar shows the navigation menu with 'API Explorer' selected. The main area is divided into two panels: 'Conditional' on the left and 'Response' on the right.

Conditional Panel:

- FROM:** +18176705441 - (817) 670-5441
- MESSAGING SERVICE SID:** (Empty field)
- BODY:** Low Water Quality Detected Due To High pH Level An
- MEDIA URL:** (Empty field)
- Optional:**
 - STATUS CALLBACK:** (Empty field)
 - APPLICATION SID:** (Empty field)

Response Panel:

- Show your Auth Token:** (Link to 'Check out the curl manpage')
- curl:**

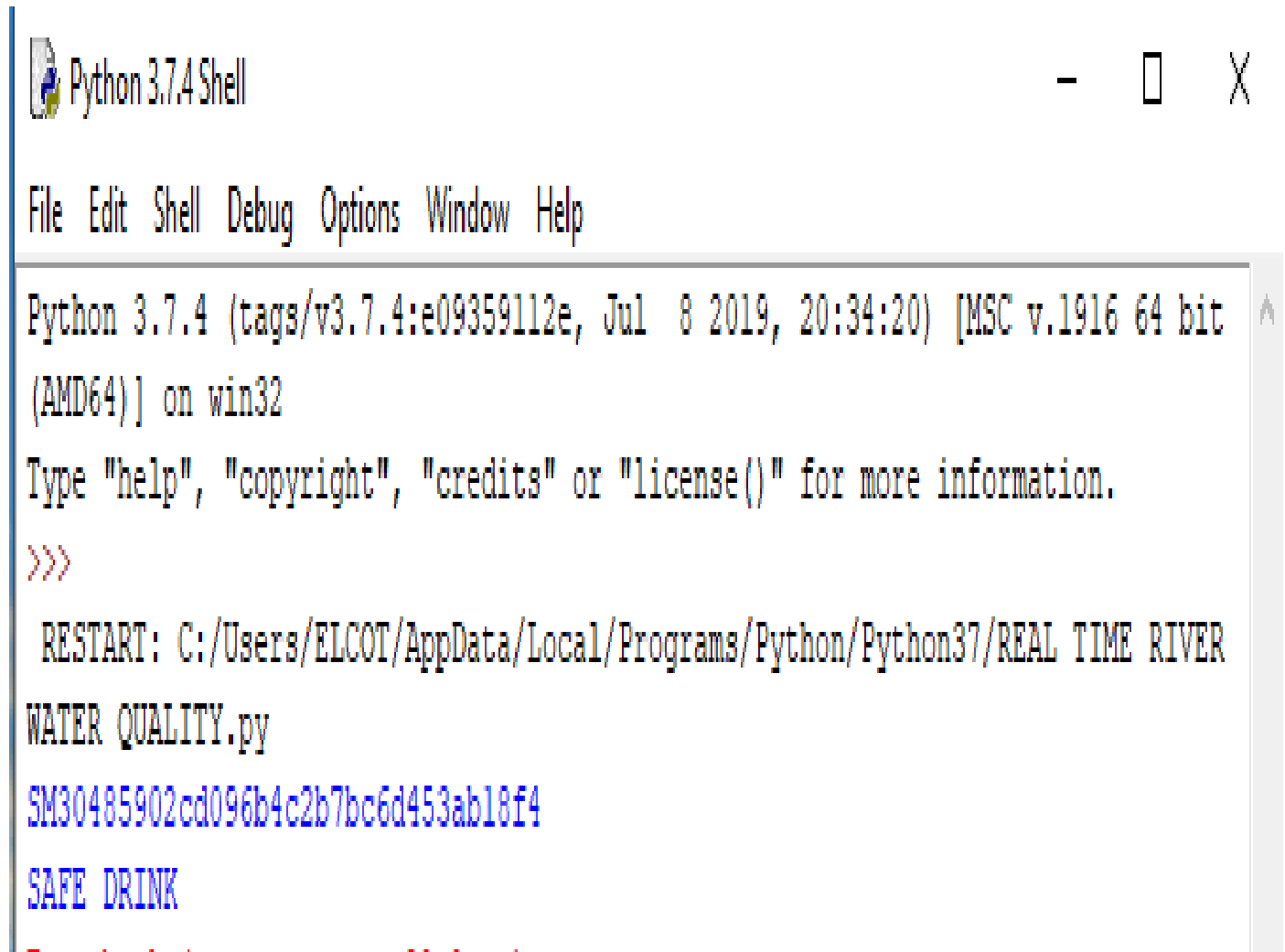
```
curl 'https://api.twilio.com/2010-04-01/Accounts/AC2f35bc97048c9abb2368f' --data-urlencode 'To=+918610214540' \ --data-urlencode 'From=+18176705441' \ --data-urlencode 'Body=Low Water Quality Detected Due To High pH Level A' -u AC2f35bc97048c9abb2368fc243b07fcbc:[AuthToken]
```
- Response:** 201 - CREATED - The request was successful. We created a new resource and the response body contains the representation.
- JSON Response:**

```
{  "body": "Sent from your Twilio trial account - Low Water Quality Detected Due To High pH Level And Humidity.Consider TURN OFF The MOT OR",  "num_segments": "1",  "direction": "outbound-api",  "from": "+18176705441",  "date_updated": "Thu, 17 Nov 2022 14:51:55 +0000",  "price": null,
```

The bottom of the image shows the Windows taskbar with the search bar and various application icons.

STEP 3 :

OUTPUT FOR PYTHON CODE:



A screenshot of a Python 3.7.4 Shell window. The title bar reads 'Python 3.7.4 Shell'. The menu bar includes 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area shows the following output: 'Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32', 'Type "help", "copyright", "credits" or "license()" for more information.', and a red prompt '>>>'. Below the prompt, the output of a script is shown: 'RESTART: C:/Users/ELCOT/AppData/Local/Programs/Python/Python37/REAL TIME RIVER WATER QUALITY.py', followed by a blue hash 'SM30485902cd096b4c2b7bc6d453ab18f4' and the text 'SAFE DRINK' in blue. A red dashed line is at the bottom.

```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/ELCOT/AppData/Local/Programs/Python/Python37/REAL TIME RIVER
WATER QUALITY.py
SM30485902cd096b4c2b7bc6d453ab18f4
SAFE DRINK
- . . . . .
```

STEP 4 :

MESSAGE SENT TO AUTHORITY WHETHER QUALITY OF WATER IS BAD OR SAFE :

