SOURCE CODE

```
#program to publish data in ibm watson iot platform
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
#Org_ID
organization = "84708c"
#Device Type
deviceType = "abcd"
#device ID
deviceId = "12345"
#Method of Authentication
authMethod = "token"
#Auth-token
authToken = "12345678"
# exception handling method
#try block
try:
deviceOptions = {"org": organization, "type": deviceType, "id":
deviceId, "auth-
method":authMethod, "auth-token":authToken}
deviceCli= ibmiotf.device.Client (deviceOptions)
#to handle the errors
except Exception as e:
print ("Caught evention connecting device: %s" % str(e))
sys.exit()
#device connection
deviceCli.connect()
#while Loop for getting the values
while True:
Ph=random.randint (6,8)
WaterTurbidity=random.randint (15,100)
salinity=random.randint (500,1000)
DissolvedOxygen=random.randint (60,130)
conductivity=random.randint (100,1200)
data = {'Ph' :
Ph,'WaterTurbidity':WaterTurbidity,'salinity':salinity,'DissolvedOxyg
en':DissolvedOxygen,'conductivity':conductivity}
#define myonpublishcallback function

def myonPublishCallback():

print ("Published Ph = %s" % Ph, "WaterTurbidity = %s %%" %
WaterTurbidity,"salinity = %s" % salinity,"DissolvedO2 = %s" %
DissolvedOxygen,"conductivity = %s" %

conductivity)
if(Ph<7.4 and salinity < 600 and DissolvedOxygen < 80 and
conductivity < 200):
if(Ph>7.4 and salinity > 900 and DissolvedOxygen > 120 and
conductivity > 1100):
```

```
print("UNSAFE, THE VALUES OF PARAMETERS ARE NOT IN
THE RANGE")

else:

print("Quality of River water is measured and its correct")

success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish =

myonPublishCallback)

if not success:

print("Not connected to IOTF")

#sleep time

time.sleep(10)

#disconnect device

deviceCli.disconnect()
```