```
“import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
import requests
import json

#Provide your IBM Watson Device Credentials Organization = “0z828r”
deviceType = “iotdevice”        #Credentials of Watson IoT sensor simulator
deviceId = “1001” authMethod = “token” authToken = “prathyusha”



# Initialize the device client.
L=0

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()

# Connect and send a datapoint “hello” with value “world” into the cloud as an event of type “greeting”
10 times deviceCli.connect()

while True:
```

```
overpass_url = http://overpass-api.de/api/interpreter
overpass_query = """
[out:json];area[name="India"];(node[place="village"](area););out;
"""

Response = requests.get(
Overpass_url,
Params={'data': overpass_query}
)

Coords = []
if response.status_code == 200:
    Data = response.json()
    Places = data.get('elements', [])
    For place in places:
        Coords.append((place['lat'], place['lon']))
    Print ("Got %s village coordinates!" % len(coords))
    Print (coords[0])
else:
    Print("Error")

I = random.randint(1,100)
L = coords[i]
#Send random gprs data to node-red to IBM Watson
Data = {"d":{ 'Latitude' : L[0], 'Longitude' : L[1]}}
#print data
def myOnPublishCallback():
    Print("Published gprs location = ", L, "to IBM Watson")
```

```
    Success = deviceCli.publishEvent(“Data”, “json”, data, qos=0, on_publish=myOnPublishCallback)
time.sleep(12)
    If not success:
        Print(“Not connected to IoTF”)
    time.sleep(1)

deviceCli.disconnect()”
```