

GPRS location.py

```
import time

import sys

import ibmiotf.application
import ibmiotf.device

import random

import requests

import json


#Provide your IBM Watson Device Credentials

organization = "aynel8"

deviceType = "iot_device"    #Credentials of Watson IoT sensor simulator

deviceId = "4016"

authMethod = "token"

authToken = "12345678"


# 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 IoT")

time.sleep(1)


deviceCli.disconnect()
```