

Develop A Web Application Using Node-RED Service

Python code for transferring latitude and longitude:

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

organization="mkgfko"
deviceType="raspberrypi"
deviceId="12345"
authMethod="token"
authToken="12345678"

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:

    #in data

    name="kowshik"

    #latitude=11.229592;

    #longitude= 78.171158;
```

```
#out data
```

```
latitude=12.7345;
```

```
longitude=13.2020;
```

```
data={'lat':latitude,'lon':longitude,'name':name}
```

```
def myOnPublishCallback():
```

```
    print("published latitude=%d" %latitude,"longitude=%d" %longitude,"to  
ibm watson")
```

```
success=deviceCli.publishEvent("lotSensor","json",data,qos=0,on_publish=myOnP  
ublishCallback)
```

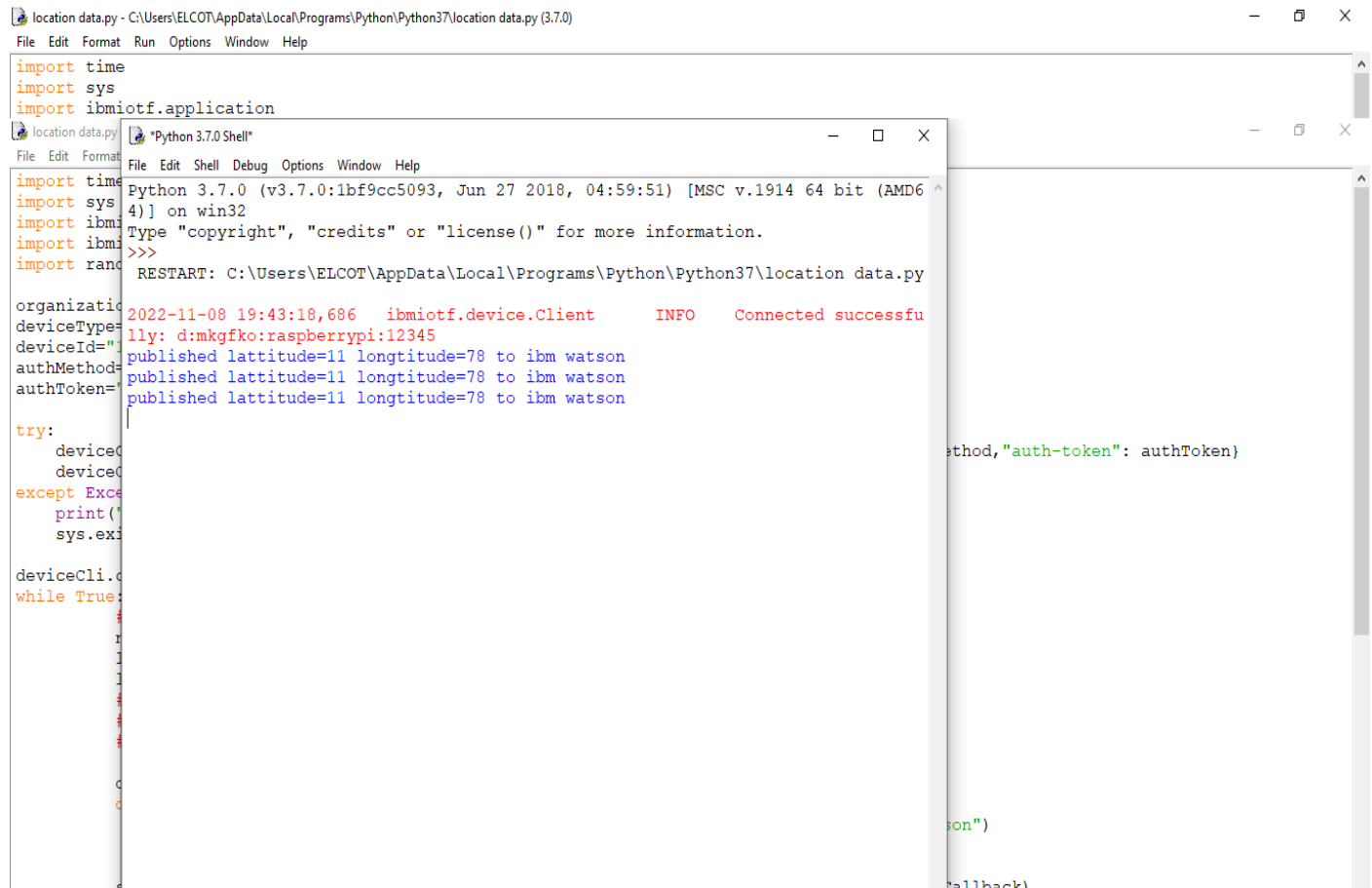
```
if not success:
```

```
    print("Not connected to IoTf")
```

```
    time.sleep(3)
```

```
deviceCli.disconnect()
```

OUTPUT:



```
location data.py - C:\Users\ELCOT\AppData\Local\Programs\Python\Python37\location data.py (3.7.0)
File Edit Format Run Options Window Help

import time
import sys
import ibmiotf.application

location data.py *Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\ELCOT\AppData\Local\Programs\Python\Python37\location data.py
2022-11-08 19:43:18,686 ibmiotf.device.Client INFO Connected successfully: d:\mkgfko:raspberrypi:12345
published latitude=11 longitude=78 to ibm watson
published latitude=11 longitude=78 to ibm watson
published latitude=11 longitude=78 to ibm watson

try:
    deviceCli.publishEvent("lotSensor","json",data,qos=0,on_publish=myOnPublishCallback)
except Exception as e:
    print(e)
    sys.exit(1)

deviceCli.disconnect()
while True:
    time.sleep(3)
```

PUBLISHED DATA IN IBM WATSON IOT PLATFORM:

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes tabs for 'Service Details - IBM Cloud', 'IBM Watson IoT Platform', and 'Node-RED'. The browser address bar shows the URL 'mkgfko.internetofthings.ibmcloud.com/dashboard/devices/browse'. The user profile 'kowschikp333777@gmail.com' with ID 'mkgfko' is visible in the top right.

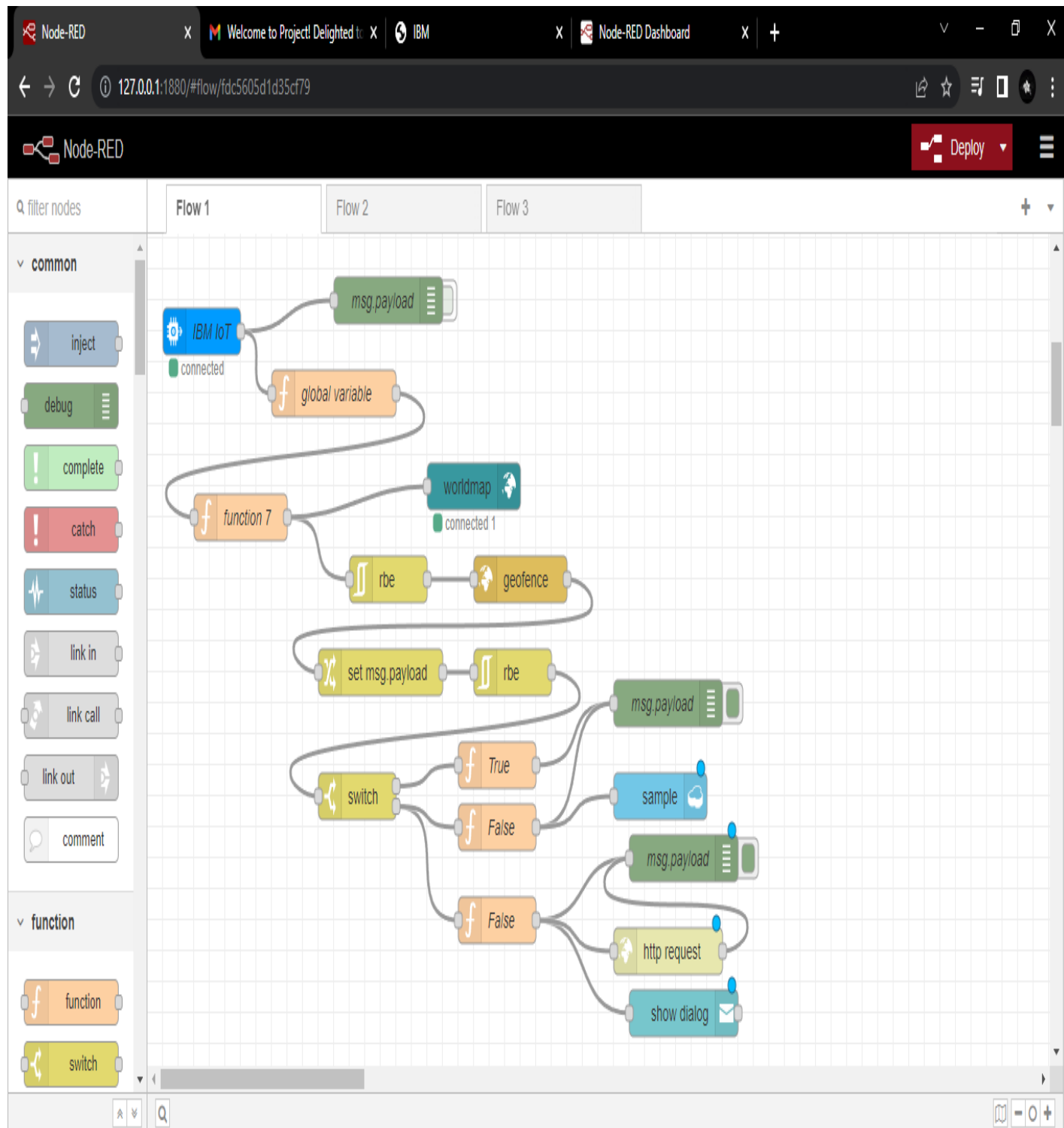
The main content area features a sidebar with icons for various functions. The central panel shows a list of devices. The selected device, ID '12345', is in a 'Connected' state and is of type 'raspberrypi'. Below the device list, the 'Recent Events' tab is active, displaying a stream of data events.

The 'Recent Events' section includes a table with the following data:

Event	Value	Format	Last Received
IotSensor	{"lat":12.7345,"lon":13.202,"name":"kowschik"}	json	a few seconds ago
IotSensor	{"lat":12.7345,"lon":13.202,"name":"kowschik"}	json	a few seconds ago
IotSensor	{"lat":12.7345,"lon":13.202,"name":"kowschik"}	json	a few seconds ago
IotSensor	{"lat":12.7345,"lon":13.202,"name":"kowschik"}	json	a few seconds ago
IotSensor	{"lat":12.7345,"lon":13.202,"name":"kowschik"}	json	a few seconds ago

Below the table, a status message indicates '0 Simulations running'.

WEB APPLICATION USING NODE-RED:



WHEN WE GIVE IN AREA LOCATION:

location data.py - C:\Users\ELCOT\AppData\Local\Programs\Python\Python37\location data.py (3.7.0)

File Edit Format Run Options Window Help

```
import time
import sys
import ibmiotf.application
```

Node-RED Welcome to Project! Delighted to... IBM Node-RED Dashboard

127.0.0.1:1880/#flow/fdc5605d1d35cf79

Node-RED Deploy

filter nodes

Flow 1 Flow 2 Flow 3

common

- inject
- debug
- complete
- catch
- status
- link in
- link call
- link out
- comment

function

- function

Flow 1

IBM IoT (connected) → msg.payload → global variable → function 7 → worldmap (connected 1) → geofence → set msg.payload → rbe → switch → True → sample → msg.payload → False → http request → show dialog

debug

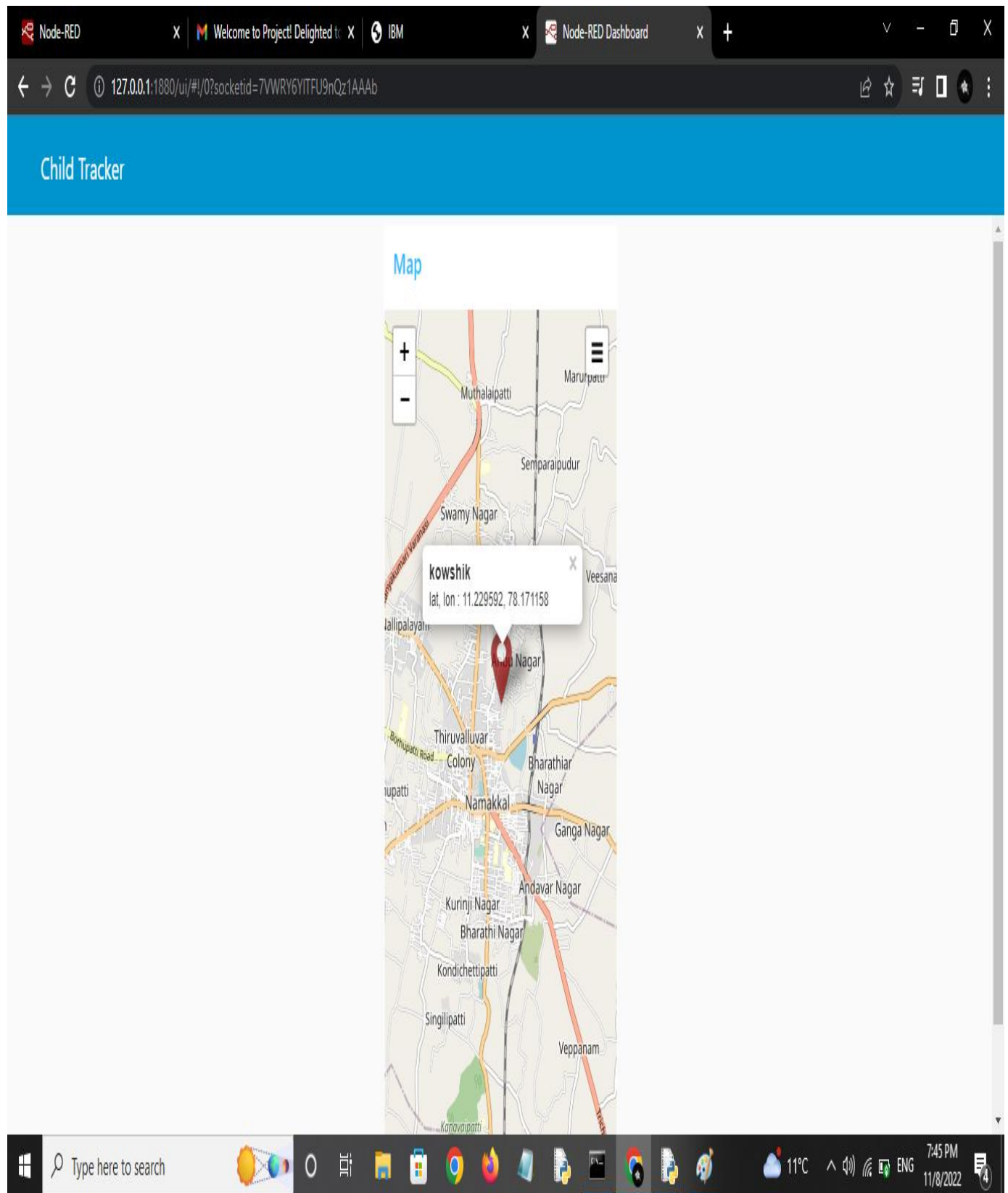
11/8/2022, 7:44:02 PM node: msg.payload

iot-2/type/raspberry/id/12345/evt/lotSensor/fmt/json : msg.payload :

Object

```
{ message: "Entry", Date and Time: "11/9/2022, 9:14:01 AM", name: "kowshik", lat: 11.229592, lon: 78.171158 }
```

OUTPUT:



WHEN WE GIVE OUT AREA LOCATION:

The image displays a Node-RED web interface in a browser window. The address bar shows the URL `127.0.0.1:1880/#flow/fdc5605d1d35cf79`. The interface includes a sidebar with node categories (common, function), a central workspace with a flow diagram, and a right-hand debug console.

Flow Diagram:

- The flow starts with an **IBM IoT** node (connected).
- It connects to a **msg.payload** node.
- Next is a **global variable** node.
- Then a **function 7** node.
- The flow then splits into two paths:
 - One path goes through a **worldmap** node (connected 1), then an **rbe** node, and a **geofence** node.
 - The other path goes through a **set msg.payload** node, then an **rbe** node.
- Both paths merge into a **switch** node.
- The **switch** node has three outputs:
 - The top output goes through a **True** function node, then a **sample** node, and a **msg.payload** node.
 - The middle output goes through a **False** function node, then a **msg.payload** node.
 - The bottom output goes through a **False** function node, then an **http request** node, and a **show dialog** node.

Debug Console:

The debug console shows three messages from the **node: msg.payload** node:

```
11/8/2022, 7:48:23 PM node: msg.payload
iot-2/type/raspberrypi/id/12345/ev/IotSensor/fmt/json : msg.payload :
Object
  { message: "Person is not in the specified.." }

11/8/2022, 7:48:23 PM node: msg.payload
iot-2/type/raspberrypi/id/12345/ev/IotSensor/fmt/json : msg.payload :
Object
  { message: "Exit", Date and Time: "11/9/2022, 9:18:23 AM", name: "kowshik", lat: 12.7345, lon: 13.202 }

11/8/2022, 7:48:24 PM node: msg.payload
iot-2/type/raspberrypi/id/12345/ev/IotSensor/fmt/json : msg.payload :
string[111]
{"status_code":999,"message":"You need to complete one transaction of 100 INR or more before using API route."}
```

OUTPUT:

