Smart Waste Management System For Metropolitan Cities

Team Id: PNT2022TMID18390

Project Development Phase

Delivery of Sprint-1

SPRINT 1

- Install Python
- Install ibmiotf using pip
- Generating Random values using python script

Source Code

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

# watson device details
organization = "5dfd8u"
devicType =  "BIN1"
deviceId = "BIN1ID"
authMethod= "token"
authToken= "123456789"

#generate random values for random variables (distance and load)
def myCommandCallback(cmd):
   global a
   print("command recieved:%s" %cmd.data['command'])
   control=cmd.data['command']
   print(control)

try:
   deviceOptions={"org": organization, "type": devicType,"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 "temp" with value integer value into the cloud as a type of
event for every 10 seconds
deviceCli.connect()
```

```
while True:

    distance= random.randint(10,70)
    loadcell= random.randint(5,15)
    data= {'dist':distance,'load':loadcell}

    if loadcell >= 13 and loadcell <= 15:
        load = "90 %"
    elif loadcell >= 8 and loadcell <= 12:
        load = "60 %"
    elif loadcell >= 5 and loadcell <= 7:
        load = "40 %"
    else:
        load = "0 %"


    if distance < 15:
        dist = "17 %"
    elif distance < 40 and distance >16:
        dist = "40 %"
    elif distance < 60 and distance > 41:
        dist = "60 %"
    else:
        dist = "90 %"


    if load == "90 %" or distance == "90 %":
        warn  = 'Alert : Dumpster poundage getting high, Time to collect :)'
    elif load == "60 %" or distance == "60 %":
        warn = 'Alert : Dumpster is above 60%'
    else:
        warn = 'Alert : No need to collect right now'

    data['alert'] = warn
    def myOnPublishCallback(lat=10.678991,long=78.177731):
        print("Location: Junction, Salem")
        print("published distance = %s " %distance,"loadcell:%s " %loadcell,"lon = %s "
%long,"lat = %s" %lat)
        print("Load %: ", load)
        print("dist %: ", dist)
        print(warn)

    time.sleep(10)

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

    if not success:
        print("not connected to ibmiot")
    time.sleep(10)

    deviceCli.commandCallback=myCommandCallback

#disconnect the device
```

```
deviceCli.disconnect()
```