

Python code

Team id : PNT2022TMID28837

IBM-Project-4533-1658733989/Untitled4.ipynb - ColaboratoryIBM Watson IoT Platform

colab.research.google.com/drive/1vbIGZiy93dzrN-g7pU5-BAPDKTRPfQ2#scrollTo=AiFxmEQYT3lq

Untitled4.ipynb

File Edit View Insert Runtime Tools Help All changes saved

[1] Successfully built ibmiotf paho-mqtt
Installing collected packages: requests-toolbelt, paho-mqtt, iso8601, ibmiotf
Successfully installed ibmiotf-0.4.0 iso8601-1.1.0 paho-mqtt-1.6.1 requests-toolbelt-0.10.1

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

# watson device details
organization = "ktym1x"
devicetype = "new"
deviceId = "09876"
authMethod = "token"
authToken = "Kamesh@2002"

#generate random values for random variables (temperature&humidity)

def myCommandCallback(cmd):
    global a
    print("command recieved:%s" %cmd.data['command'])
    control=cmd.data['command']
    print(control)

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 "temp" with value integer value into the cloud as a type of event for every 10 seconds
```

2m 41s completed at 4:11 PM

Comment Share

Enable browser notifications in Settings to get alerts when executions complete

OK No thanks

Type here to search

29°C Haze 4:11 PM 11/18/2022

The screenshot shows a Jupyter Notebook in a web browser. The notebook is titled "Untitled4.ipynb" and contains the following Python code:

```
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 < 4 and loadcell > 7:
    load = "40 %"
else:
    load = "0 %"

if distance < 15:
    dist = 'Risk warning:' 'Dumpster poundage getting high, Time to collect :) 90 %'

elif distance < 40 and distance >16:
    dist = 'Risk warning:' 'dumpster is above 60%'

elif distance < 60 and distance > 41:
    dist = 'Risk warning:' '40 %'
else:
    dist = 'Risk warning:' '17 %'

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 '
def myOnPublishCallback(lat=10.678991,lon=78.177731):
```

The status bar at the bottom indicates "2m 41s completed at 4:11 PM". A notification bubble in the top right corner says "Enable browser notifications in Settings to get alerts when executions complete" with "OK" and "No thanks" buttons.

IBM Project-4533-1658733989/Untitled4.ipynb - ColaboratoryIBM Watson IoT Platform

colab.research.google.com/drive/1vblGZiy93dzrN-g7pU5-BAPDKTRPfq2#scrollTo=AiFxmEoyT3fq

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

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

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


if not success:
    print("not connected to ibmiot")
    time.sleep(30)


deviceCli.commandCallback=myCommandCallback
#disconnect the device
deviceCli.disconnect()
```

2022-11-18 10:38:44,079 ibmiotf.device.Client INFO Connected successfully: d:ktymLx:new:09876

INFO:ibmiotf.device.Client:Connected successfully: d:ktymLx:new:09876

chennai, ,manimangalam

published distance = 19 loadcell:15 lon = 78.177731 lat = 10.678991

0 %

Risk warning:dumpster is above 60%

alert :No need to collect right now

chennai, ,manimangalam

published distance = 19 loadcell:15 lon = 78.177731 lat = 10.678991

0 %

Risk warning:dumpster is above 60%

alert :No need to collect right now

chennai, ,manimangalam

published distance = 48 loadcell:11 lon = 78.177731 lat = 10.678991

0 %

Risk warning:40 %

alert :No need to collect right now

chennai, ,manimangalam

published distance = 48 loadcell:11 lon = 78.177731 lat = 10.678991

0 %

Risk warning:40 %

alert :No need to collect right now

2m 41s completed at 4:11 PM

29°C Haze 4:12 PM 11/18/2022

Comment Share

Enable browser notifications in Settings to get alerts when executions complete

OK No thanks

IBM Watson IoT Platform

Search by Device ID

Device Simulator ☒

Device ID **Status** **Device Type** **Class ID** **Date Added** **Descriptive Location**

Event	Value	Format	Last Received
IoTSensor	{"dist":48,"load":14}	json	a few seconds ago
IoTSensor	{"type":"Buffer","data":[34,97,108,101,114,116,...]}	json	a few seconds ago
event	{"Alert distance":94.5}	json	a few seconds ago
IoTSensor	{"dist":68,"load":5}	json	a few seconds ago
IoTSensor	{"type":"Buffer","data":[34,97,108,101,114,116,...]}	json	a few seconds ago

Items per page 50 | 1-1 of 1 item

1 Simulation running

Snip & Sketch

Snip saved to clipboard
Select here to mark up and share the image

Code:

```
import requests
```

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

# watson device details

organization = "ktymlx"
devicType = "new"
deviceId = "09876"
authMethod= "token"
authToken= "Kamesh@2002"

#generate random values for randomo variables (temperature&humidity)

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": device
Id,"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 clo
ud 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 < 4 and loadcell > 7:

```

```

        load = "40 %"
    else:
        load = "0 %"

    if distance < 15:
        dist = 'Risk warning:' 'Dumpster poundage getting high, Time to
collect :) 90 %'

    elif distance < 40 and distance >16:
        dist = 'Risk warning:' 'dumpster is above 60%'

    elif distance < 60 and distance > 41:
        dist = 'Risk warning:' '40 %'
    else:
        dist = 'Risk warning:' '17 %'

    if load == "90 %" or distance == "90 %":
        warn = 'alert :' ' Dumpster poundage getting high, Time to coll
ect :)'

    elif load == "60 %" or distance == "60 %":

```



```

        warn = 'alert :' 'dumpster is above 60%'
    else :
        warn = 'alert :' 'No need to collect right now '
    def myOnPublishCallback(lat=10.678991,long=78.177731):
        print("chennai, ,manimangalam")
        print("published distance = %s " %distance,"loadcell:%s " %loadcel
1,"lon = %s " %long,"lat = %s" %lat)
        print(load)
        print(dist)
        print(warn)

    time.sleep(10)

    success=deviceCli.publishEvent ("IoTSensor","json",warn,qos=0,on_publi
sh= myOnPublishCallback)

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

    if not success:
        print("not connected to ibmiot")
        time.sleep(30)

```

```
deviceCli.commandCallback=myCommandCallback  
#disconnect the device  
deviceCli.disconnect()
```