Hello World Device Wrapper

The following is a hello-world python process that writes a message to i3 core information fabric as an ‘event’ data type from the device ‘Gerard’. As an process that generates messages for submittal to the i3 information fabric, this type of an process would be classified as a device wrapper. If an application has been given access to the event messages coming from ‘gerard’, this message would be delivered to the application.

```
from i3_broker_device import I3BrokerDevice
from i3_broker_connection import I3BrokerConnection


BROKER_HOST = '3.16.219.147'  #IP of an i3 core node
BROKER_PORT = 1883            #target port on the i3 core node
DEVICE_NAME = 'gerard'        #the device name associated with this message
DEVICE_PASSWORD = '123456'    #password to authenticate this device
DEVICE_TOPIC = 'event'        #information data type

i3_broker_device = I3BrokerDevice(BROKER_HOST, BROKER_PORT, DEVICE_NAME,
DEVICE_PASSWORD)

i3_broker_device.connect()

message="**************** Hello World, i3 reporting for duty ******************"

i3_broker_device.publish(DEVICE_TOPIC,message)time.sleep(1)
```

The BROKER_HOST and BROKER_PORT parameters identify the location of the i3 core software and it messaging port. The DEVICE_NAME and DEVICE_PASSWORD are used to authenticate this device wrapper to the i3 system. These parameters are aligned with the device name and device password defined in the i3 system under the Spaces device submenu. The DEVICE_TOPIC parameter identifies the type of message this application is sending to the i3 system can route the message to all applications that have been authorized for access. Device topics are associated with data types defined in i3’s administrative menu.

A slightly more complicated version of the ‘hello-world’ application serves to demonstrate a python application that has the ability to write to two different data types.

```
from i3_broker_device import I3BrokerDevice
from i3_broker_connection import I3BrokerConnection


BROKER_HOST = '3.16.219.147'      #IP of the staging node
BROKER_PORT = 1883                #target port on the i3 core node
DEVICE_NAME = 'gerard'            #the device name associated with this message
DEVICE_PASSWORD = '123456'        #password to authenticate this device
DEVICE_TOPIC_EVENT = 'event'      #information data type number one
DEVICE_TOPIC_LOCATION='locate'    #information data type number two

i3_broker_device = I3BrokerDevice(BROKER_HOST, BROKER_PORT, DEVICE_NAME,
DEVICE_PASSWORD)

i3_broker_device.connect()

message1="**************** Hello World, i3 reporting for duty ******************"
message2="*** Device at lat 25.05 long -118.24 ***"

i3_broker_device.publish(DEVICE_TOPIC_EVENT,message1)
i3_broker_device.publish(DEVICE_TOPIC_LOCATION,message2)
```

In the case where a device wrapper is writing to two data types, if the application has been approved to receive both information messages, then both data types will be delivered to the application. However, if the application has only been approved to receive one data type from the device ‘Gerard’, the application will receive only that one data type of the two.

The i3 system can not only gather information from devices, it can also be used to gather data from websites that produce data as well. The following device wrapper gets current weather information from api.openweathermap.org. The information that is returned in response to the query is put in a message and set to the i3 core. In this case, the device proxy is represented in i3 as a device wrapper named ‘LA-Weather’ that provides information to the ‘weather’ data type.

As programmed, the device wrapper will query the website for the latest weather every 30 seconds for two hours.

```
# This python i3 wrapper pulls LA weather data from the openweather site
# URL http://api.openweathermap.org/data/2.5/weather?q=Los
Angeles&appid=35446fdadf0feadbcfed6383da248515
#
import json
import time
import requests
import sys

from i3_broker_device import I3BrokerDevice

def get_message():
    data = requests.get(DEVICE_WEB_SERVER, PARMS)  #
    json_data = data.json()
    return json.dumps(json_data)

BROKER_HOST = '3.16.219.147'  # IP Address of I3 node
BROKER_PORT = 1883            # incoming Port for I3 node
DEVICE_NAME = 'LA-Weather'    # should match i3 device name
DEVICE_PASSWORD = '123456'    # should match i3 device authentication pwd
DEVICE_TOPIC1 = 'weather'     # should match i3 datatype

FREQUENCY  = 1*60/2           # FREQUENCY how many secs between queries
DURATION = 2*60*60            # DURATION is how long the program runs (secs)
STOP = time.time()+DURATION

DEVICE_WEB_SERVER = 'https://api.openweathermap.org/data/2.5/weather/'
PARMS={'q':'Los Angeles','appid':'xxxxxxxxxxxx','units':'imperial'}
# replace xxxxxxxxxxxx with an application key obtained from openweathermap.org

i3_broker_device = I3BrokerDevice(BROKER_HOST, BROKER_PORT, DEVICE_NAME,
DEVICE_PASSWORD)
i3_broker_device.connect()

while time.time() < STOP :
    message = get_message()
    i3_broker_device.publish(DEVICE_TOPIC1, message)
    time.sleep(FREQUENCY)

i3_broker_device.disconnect()
```