

ASSIGNMENT 4 by Sanskar Vidyarthi

Develop a mobile application that takes the user input and sends it to IoT device (python code). print the received data in python shell.

Keep a text box to accept the user input.integrate a submit button.

whenever user enters the text input in text box and clicks the button the data should be sent to IBM cloud using URL(HTTP API).

Code-

```
import wiotp.sdk.device
import time
import random

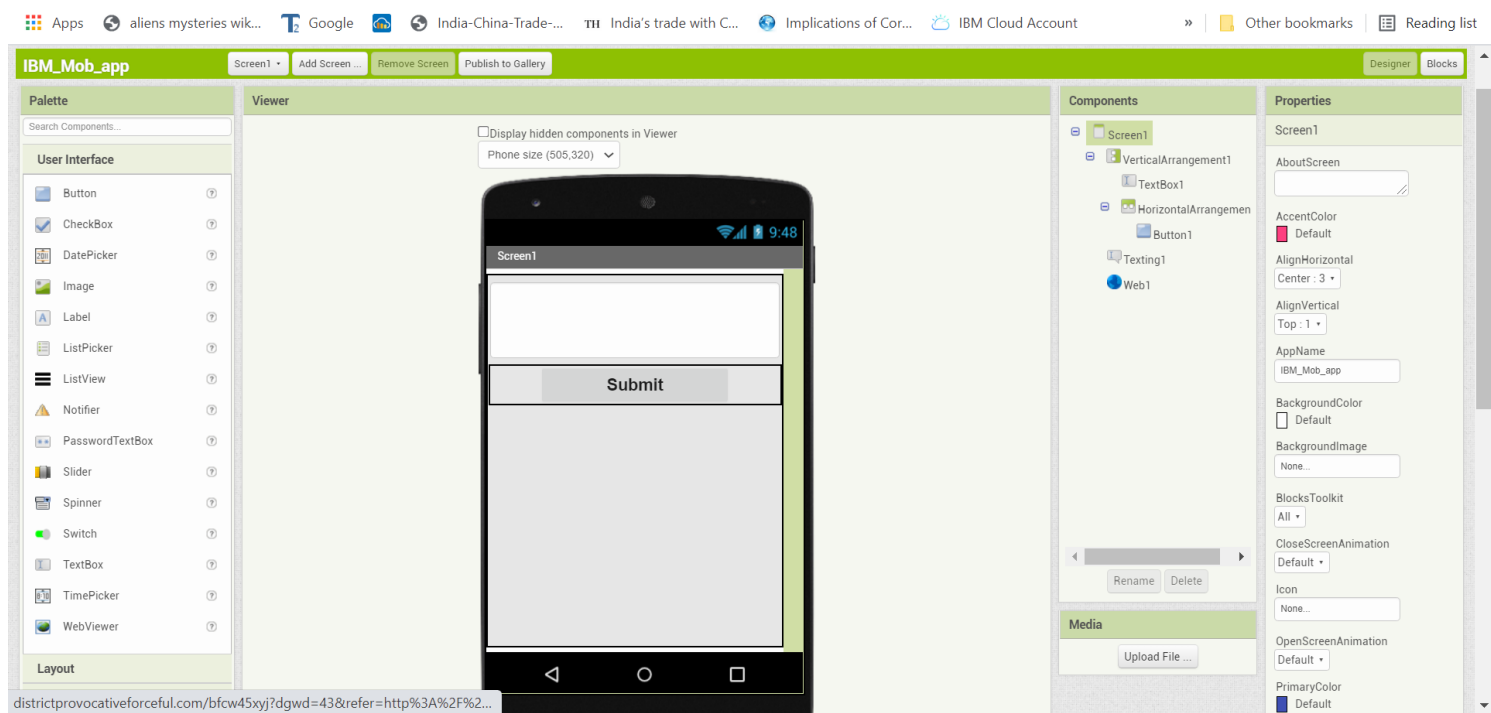
myConfig = {
    "identity": {
        "orgId": "3ma3me",
        "typeId": "IoTDevice",
        "deviceId": "12345"
    },
    "auth": {
        "token": "Logan1716"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']

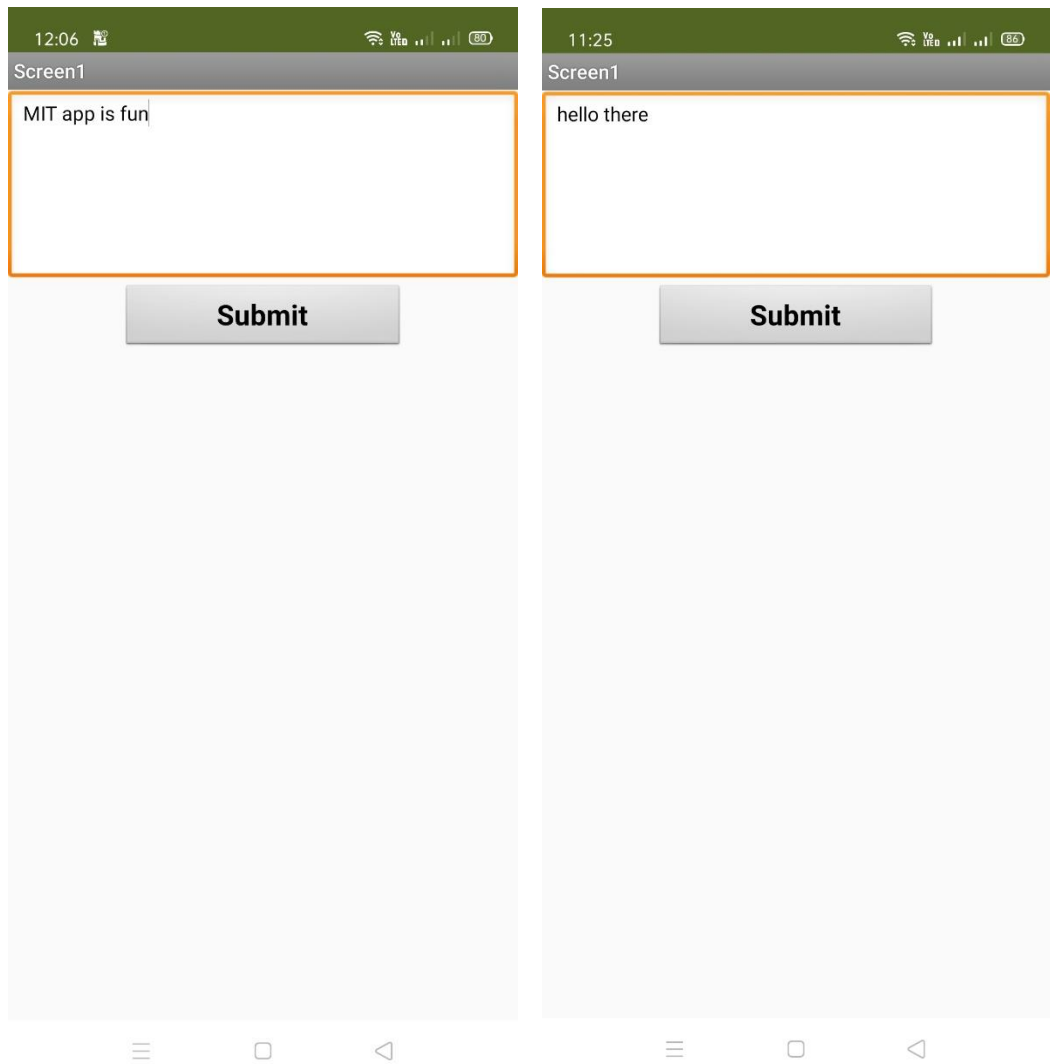
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```

MIT App inventor screen



Messages sent through the app



Message received in Python IDLE/Jupyter notebook and Node red

Appsaliens mysteries wik...India-China-Trade...India's trade with C...Implications of Cor...IBM Cloud Account»Other bookmarksReading list

jupyter MIT AppLast Checkpoint: 26 minutes ago (autosaved)

Logout

FileEditViewInsertCellKernelWidgetsHelpNotebook savedTrustedPython 3

import random

myConfig = {

"identity": {

"orgId": "3ma3me",

"typeId": "IoTDevice",

"deviceId": "12345"

},

"auth": {

"token": "Logan1716"

}

}

def myCommandCallback(cmd):

print("Message received from IBM IoT Platform: %s" % cmd.data['command'])

m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)

client.connect()

while True:

client.commandCallback = myCommandCallback

time.sleep(2)

client.disconnect()

2021-07-19 00:04:03,489 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:3ma3me:IoTDevice:12345

Message received from IBM IoT Platform: hello there

Message received from IBM IoT Platform: MIT app is fun

The screenshot shows the Node-RED web interface. On the left, there's a sidebar with a 'common' section containing nodes like inject, debug, complete, catch, status, link in, link out, and comment. Below that is a 'function' section with a function node. The main workspace displays a flow diagram with a 'Send' node connected to an 'IBM IoT' node and a 'msg.payload' node. The 'IBM IoT' node is connected to an 'http' node. The right sidebar shows a 'debug' section with a log of messages, including a 'hello there' command and a 'MIT app is fun' command.