```
/*
 Name:      BFSketch1.ino
 Created:   12/26/2017 10:47:23 AM
 Author:    brian
*/



#include <ArduinoJson.h>
#include <ESP8266WiFi.h>
#include <WebSocketsClient.h>
#include <ESP8266WiFiMulti.h>
#include <Hash.h>


const char* ssid = "LinksysF";
const char* password = "1234567890";
char host[] = "bf-skill-set.herokuapp.com/";

const int relayPin = BUILTIN_LED;
int pingCount = 0;

int port = 80;
char path[] = "/ws";

ESP8266WiFiMulti WiFiMulti;
WebSocketsClient webSocket;

DynamicJsonBuffer jsonBuffer;
String currState, oldState, message;

void webSocketEvent(WStype_t type, uint8_t * payload, size_t length)
{


    switch (type) {
    case WStype_DISCONNECTED:
        //USE_SERIAL.printf("[WSc] Disconnected!\n");
        Serial.println("Disconnected! ");
        break;

    case WStype_CONNECTED:
    {
        Serial.println("Connected! ");
        // send message to server when Connected
        webSocket.sendTXT("Connected");
    }
    break;

    case WStype_TEXT:
        Serial.println("Got data");
```

```
54          //Serial.println("looping...");
55          //digitalWrite(trigPin, HIGH);
56          //delayMicroseconds(10);
57          //digitalWrite(trigPin, LOW);
58          //duration = pulseIn(echoPin, HIGH);
59          //distance = (duration / 2) / 74; //Inches
60          //distance = (duration / 2) / 29.1; //centimeter
61
62          if (BUILTIN_LED ) {
63              Serial.print(BUILTIN_LED);
64              currState = "open";
65              Serial.println(currState);
66          }
67          else {
68              Serial.println(currState);
69              currState = "close";
70          }
71
72
73
74          processWebScoketRequest((char*)payload);
75
76          break;
77
78      case WStype_BIN:
79
80          hexdump(payload, length);
81          Serial.print("Got bin");
82          // send data to server
83          // webSocket.sendBIN(payload, length);
84          break;
85      }
86
87  }
88
89  // the setup function runs once when you press reset or power the board
90  void setup() {
91
92      Serial.begin(115200);
93      // Setup Relay
94      pinMode(relayPin, OUTPUT);
95
96      for (uint8_t t = 4; t > 0; t--) {
97          delay(1000);
98      }
99      Serial.println();
100     Serial.println();
101     Serial.print("Connecting to ");
102
103     //Serial.println(ssid);
104     WiFiMulti.addAP(ssid, password);
105
106     //WiFi.disconnect();
```

```
107     while (WiFiMulti.run() != WL_CONNECTED) {
108         Serial.print(".");
109         delay(1000);
110     }
111
112     Serial.println("Connected to wi-fi");
113     webSocket.begin(host, port, path);
114     webSocket.onEvent(webSocketEvent);
115
116 }
117
118 // the loop function runs over and over again until power down or reset
119 void loop() {
120
121     webSocket.loop();
122     delay(100);
123     // make sure after every 40 seconds send a ping to Heroku
124     //so it does not terminate the websocket connection
125     //This is to keep the conncetion alive between ESP and Heroku
126     if (pingCount > 20) {
127         pingCount = 0;
128         webSocket.sendTXT("\"heartbeat\":\"keepalive\"");
129     }
130     else {
131         pingCount += 1;
132     }
133
134
135 }
136
137 void processWebScoketRequest(String data) {
138     String jsonResponse = "{\"version\": \"1.0\",\"sessionAttributes\":
        {},\"response\": {\"outputSpeech\": {\"type\": \"PlainText\",\"text
        \": \"<text>\"},\"shouldEndSession\": true}}";
139     JsonObject& req = jsonBuffer.parseObject(data);
140
141     String instance = req["Instance"];
142     String state = req["Light_Command"];
143     String query = req["Light_Question"];
144     String message = "{\"event\": \"OK\"}";
145
146     Serial.println("Data2-->" + data);
147     Serial.println("State-->" + state);
148
149     if (query == "?") { //if command then execute
150         Serial.println("Recieved query!");
151         if (currState == "on") {
152             message = "on";  //This could be On
153         }
154         else {
155             message = "off"; //This clould be Off
156         }
157         jsonResponse.replace("<text>", "Lights" + instance + " are " +
```

```
            message);
158         webSocket.sendTXT(jsonResponse);
159
160     }
161     else if (query == "cmd") { //if query check state
162         Serial.println("Recieved command!");
163         if (state != currState) {
164             if (currState == "off") {
165                 message = "Switching on"; // Switching On
166                 digitalWrite(relayPin, HIGH); //Switch On
167             }
168             else {
169                 message = "Switching off"; //Switching Off
170                 digitalWrite(relayPin, LOW); //Switch Off
171             }
172             //digitalWrite(relayPin, HIGH); //Switch on
173             //delay(1000);
174             //digitalWrite(relayPin1, LOW);
175         }
176         else {
177             if (currState == "off") {
178                 message = "already off"; //This could say already Off
179             }
180             else {
181                 message = "already on"; //This Could say already On
182             }
183         }
184         jsonResponse.replace("<text>", "Light " + instance + " are " +
            message);
185         webSocket.sendTXT(jsonResponse);
186
187
188     }
189     else {//can not recognized the command
190         Serial.println("Command is not recognized!");
191         jsonResponse.replace("<text>", "Command is not recognized by
            garage door Alexa skill");
192         webSocket.sendTXT(jsonResponse);
193     }
194     Serial.print("Sending response back");
195     Serial.println(jsonResponse);
196     // send message to server
197     webSocket.sendTXT(jsonResponse);
198 }
199
```