1. The following are the full code after page 13 in the book.

```
#include <FirebaseArduino.h>
#include <ESP8266WiFi.h>

// Set these to run example.
#define FIREBASE_HOST "example.firebaseio.com"
#define FIREBASE_AUTH "token_or_secret"
#define WIFI_SSID "your_ssid"
#define WIFI_PASSWORD "your_password"


//Global declaration and initialization of variables
const int LED1 = 4;     //D2 GPIO4
const int LED2 = 14;    //D5 GPIO14
const int LED3 = 15;    //D8 GPIO15
const int button = 16;  //D0 GPIO16
int temp = 0;           //initialize temporary variable


void setup() {
  Serial.begin(115200); // Communication at 115200 Bd with Serial
Monitor

//initialize digital pin as an input/output
  pinMode(LED1,OUTPUT);
  pinMode(LED2,OUTPUT);
  pinMode(LED3,OUTPUT);
  pinMode(button,INPUT);

  // connect to wifi.
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  delay(1000);

}

void loop() {

  // get value from firebase
  String led1_status=(Firebase.getString("/LED1/status"));
  //Serial.println(led1_status);
  String led2_status=(Firebase.getString("/LED2/status"));
  //Serial.println(led2_status);
  String led3_status=(Firebase.getString("/LED3/status"));
  //Serial.println(led3_status);
```

```
  if(led1_status=="ON"){
    digitalWrite(LED1,HIGH);
  }else{
    digitalWrite(LED1,LOW);
  }
  if(led2_status=="ON"){
    digitalWrite(LED2,HIGH);
  }else{
    digitalWrite(LED2,LOW);
  }
  if(led3_status=="ON"){
    digitalWrite(LED3,HIGH);
  }else{
    digitalWrite(LED3,LOW);
  }

  temp = digitalRead(button);
    if(temp==HIGH){
        // set value of PushButton to HIGH
        Firebase.setString("/PushButton/status", "HIGH");
        Serial.println("HIGH");

        // handle error
        if (Firebase.failed()) {
            Serial.print("setting /BUTTON/status failed:");
            Serial.println(Firebase.error());
            return;
        }
      }
    else{
      // set value of PushButton to LOW
        Firebase.setString("/PushButton/status", "LOW");
        Serial.println("LOW");
        // handle error
        if (Firebase.failed()) {
            Serial.print("setting /BUTTON/status failed:");
            Serial.println(Firebase.error());
            return;
        }
      }
   }
```

Due to Firebase no longer accept FCM legacy starting early September 2019, the code listed in page 48 and beyond need to be amended to the following steps:

13. Open ***ControlActivity.java.*** After the ***onCreate(…)*** method, create a ***getCurrentToken()*** method as follows:

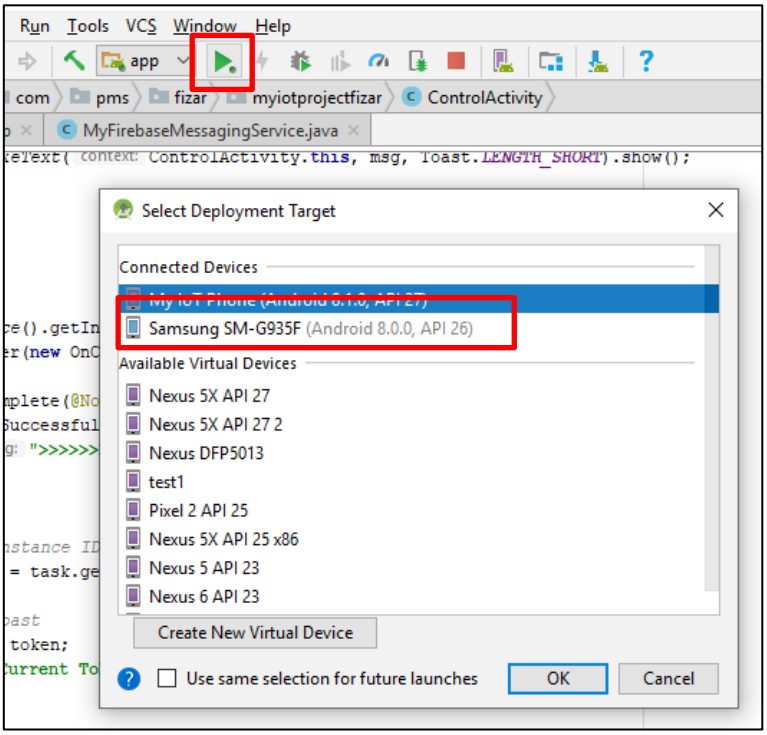
```
private void getCurrentToken() {

    FirebaseInstanceId.getInstance().getInstanceId()
            .addOnCompleteListener(new OnCompleteListener<InstanceIdResult>() {
                @Override
                public void onComplete(@NonNull Task<InstanceIdResult> task) {
                    if (!task.isSuccessful()) {
                        Log.w(">>>>>>>", "getInstanceId failed",
task.getException());
                        return;
                    }
```

```
                    // Get new Instance ID token
                    String token = task.getResult().getToken();

                    // Log and toast
                    String msg = token;
                    Log.d(">>>>>>>>>>", msg);
                }
            });
}
```

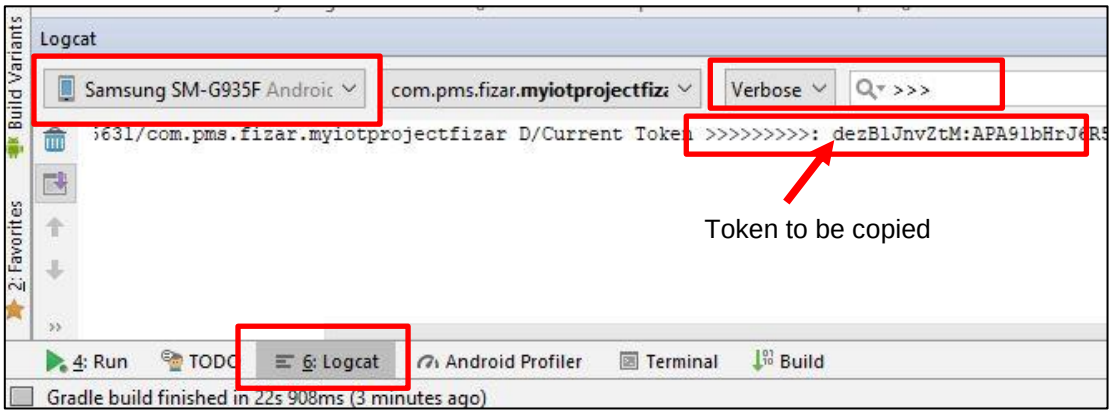
14. Call the ***getCurrentToken()*** method inside the ***onCreate()*** method.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_control);

    sw1 = findViewById(R.id.switch1);
    tbLed2 = findViewById(R.id.tbLed2);
    tvSwitchStatus = findViewById(R.id.tvSwitchStatus);
    rg = findViewById(R.id.radioGroup1);

    getCurrentToken();
    //subscribeToTopic();  no longer used
```

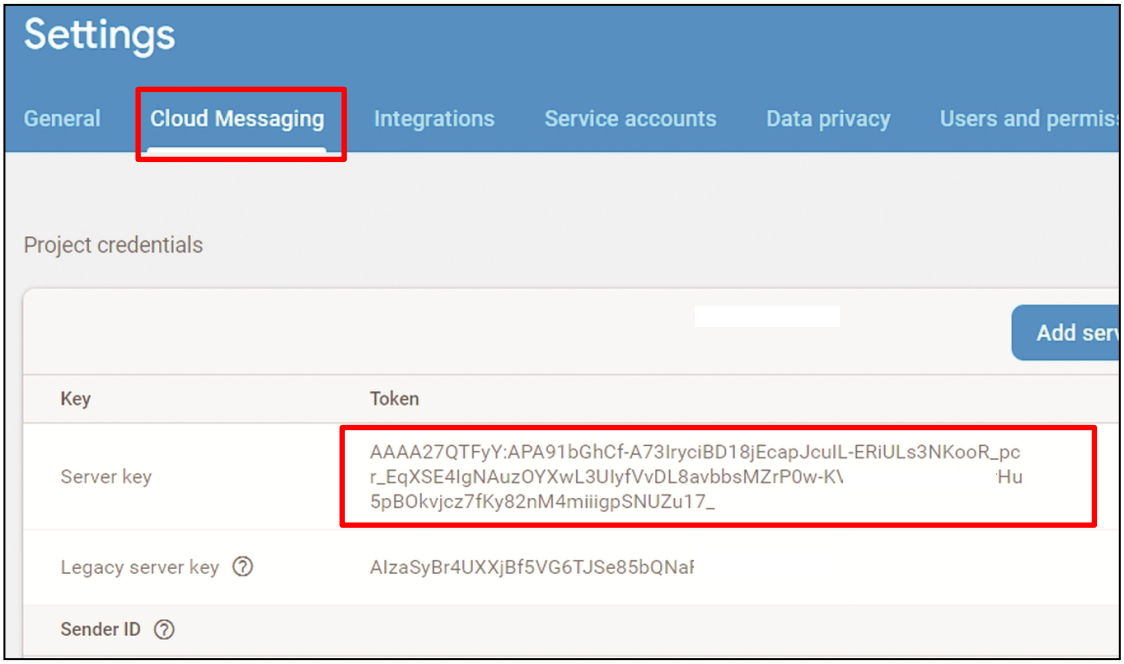
15. Click ***Build > Rebuild Project***.
16. **Click Run > Run app** or Click on the ***Run icon***
17. Connect your Android phone to your computer via USB cable. If you are using an emulator, select the virtual device. Click **OK**.



18. Open Android Studio logcat to copy current device token of Android phone / emulator. For easier search, select the output type as Verbose and search ***'>>>'.*** The android phone / emulator unique token will be displayed. Copy the token into the ***MyFirebase.ino*** file.




19. Go to your Firebase console.
20. Click on ***Cloud Messaging.*** Copy the ***Server key.***



## 1. FCM Setup in NodeMCU

1. Launch your Arduino IDE. Click ***File > Open > MyFirebase.ino***
2. Add ***#include <WiFiClientSecure.h>***


3. After the setup() function, add the following sendDataToFirebase() function.

```
void sendDataToFirebase() {  //new changes
  WiFiClientSecure client;
  String data = "{";
  data = data + "\"to\": \"" + reg + "\",";
  data = data + "\"notification\": {";
  data = data + "\"body\": \"Some pressed your doorbell.\",";
  data = data + "\"title\" : \"Info\" ";
  data = data + "} }";
  Serial.println("Send data...");
  if (client.connect("fcm.googleapis.com", 443)) {
    Serial.println("Connected to the server..");
    client.println("POST /fcm/send HTTP/1.1");
    client.println("Authorization: key=" + serve + "");
    client.println("Content-Type: application/json");
    client.println("Host: fcm.googleapis.com");
    client.print("Content-Length: ");
    client.println(data.length());
    client.print("\n");
    client.print(data);
  }
  Serial.println("Data sent...Reading response..");
  while (client.available()) {
    char c = client.read();
    Serial.print(c);
  }
```

```
  Serial.println("Finished!");
  client.flush();
  client.stop();
}
```

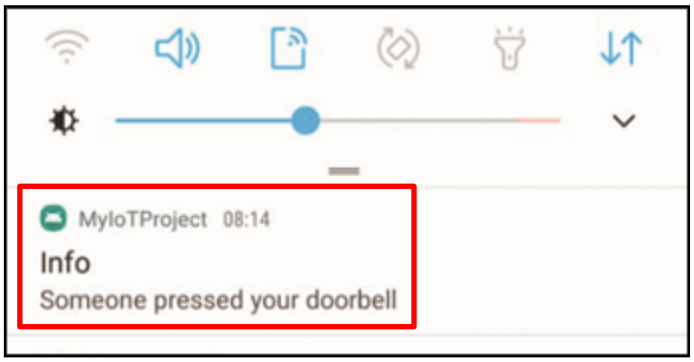
4. Call the  function as below.

```
temp = digitalRead(button);
  if(temp==HIGH){
      // set value of PushButton to HIGH
      Firebase.setString("/PushButton/status", "HIGH");
      Serial.println("HIGH");

      //send notification
      sendDataToFirebase();

      // handle error
      if (Firebase.failed()) {
          Serial.print("setting /BUTTON/status failed:");
          Serial.println(Firebase.error());
          return;
      }
   }
```

5. Add the android phone / emulator device token from ***Step 18*** to a string variable named **reg**.
6. Add the ***server key*** copied from firebase from ***Step 20*** to a string variable named **serve**.

```
#define WIFI_SSID "your_ssid"
#define WIFI_PASSWORD "your_password"

String serve = "AAAAQIugd3c:APA91bFrN5p2c68we00Uff7bbvuUcx7zTh
String reg = "dezBlJnvZtM:APA91bHrJ6R5V8NClBrSCokX9j8l1eXWGo4Zx
```

7. ***Upload*** your sketch to NodeMCU.
8. Launch your android app.
9. You should be able to receive a push notification whenever the push button at NodeMCU is pressed.




Congratulations! You have completed a simple and basic project of Internet of Things using the NodeMCU board, Android Studio, Firebase Database and have successfully control your IoT device via your android app.

**Note**: If after some time your project no longer can connect to Firebase and keeps getting Firebase error, you need to check the Firebase fingerprint as sometime Google will update the fingerprint. To find out the current valid fingerprint for your firebase project, you can use the website *https://www.grc.com/fingerprints.htm* and type in ***<your_firebase_url>*** (e.g: ***myfirebase4iot.firebaseio.com***) and click the Fingerprint Site button to get the new fingerprint. Then go to **"Arduino/libraries/firebase-arduino-master/src/FirebaseHttpClient.h"** and change the line of kFirebaseFingerprint[]="xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx" to the new fingerprint.

You can download the relevant code for NodeMCU and android app from this github link: ***rafizahabrahman/Firebase_4_IoT***