Intro to Firebase Database

ECE 196
Ravi Patel

Agenda

- Firebase Setup
- Understanding Realtime Database
- Integrating Firebase w/ Raspberry Pi
- Integrating Firebase w/ Android Studio

Firebase Setup

Firebase Setup

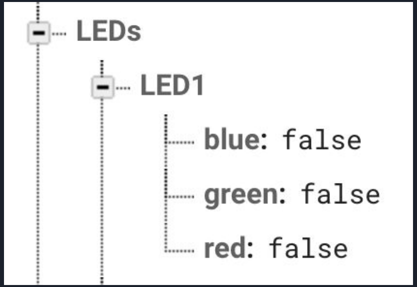
- Refer to the following link to get started with Firebase Database
  - https://docs.google.com/document/d/17ndeuLeW8Z4eBbcEDmc3TD7sImaCpgigRALZRzLwqvo/edit?usp=sharing
- Refer to the following for the pricing options
  - https://firebase.google.com/pricing

Real-time Database

Real-Time Database vs Cloud Firestore

Real-Time Database

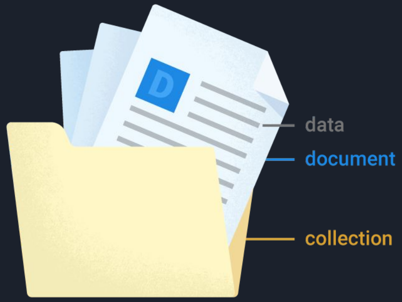
- NoSQL Database
- JSON Structure



```
LEDs
    LED1
        blue: false
        green: false
        red: false
```

Cloud Firestore

- NoSQL Database
- Stores data in documents
  - Arranged in collections



Firebase with Raspberry Pi

1. Generate the Firebase-Admin Key

Refer to the following link to generate the Firebase-Admin key and save the key on the **Raspberry Pi**.

2. Authenticating Credentials

```
import firebase_admin

from firebase_admin import credentials

cred = credentials.Certificate(  “PATH_TO_SERVICE_ACCOUNT_FILE.json”  )

firebase_admin.initialize_app( cred, { “databaseURL”:  “LINK_TO_DATABASE_URL” } )
```

NOTE: replace the **PATH_TO_SERVICE_ACCOUNT_FILE** with the appropriate path

NOTE: replace the **LINK_TO_DATABASE_URL** with the appropriate URL

NOTE: Ues the following link to refer to the official documentation

3. Creating Database Reference

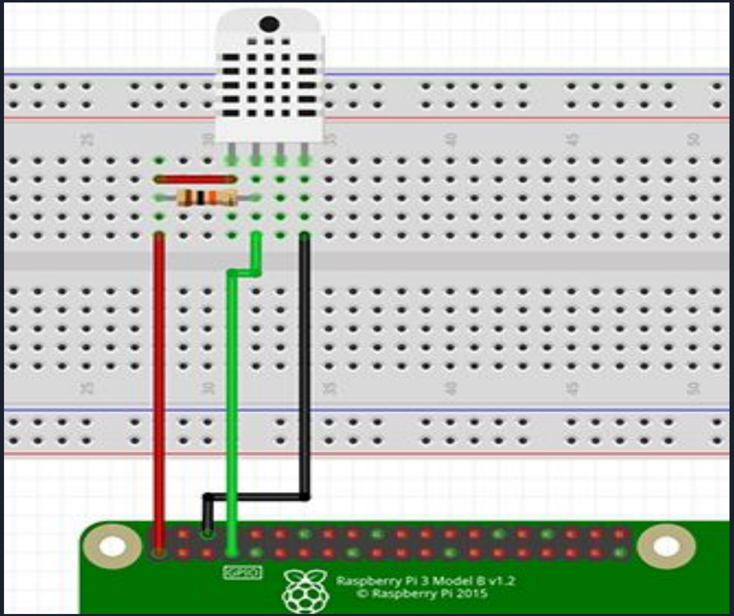
```
from firebase_admin import db

db_ref = db.reference( “PATH_TO_REFERENCE” )

db_ref.set( VALUE )

data = db_ref.get()
```

NOTE: replace the **PATH_TO_REFERENCE** with the appropriate path

NOTE: replace **VALUE** with the appropriate value

Challenge 1 -- Revisiting DHT22

Modify the DHT22 Sensor code to read temperature and humidity value every 10 sec and save the value to the firebase database.



Challenge 1 -- Solution

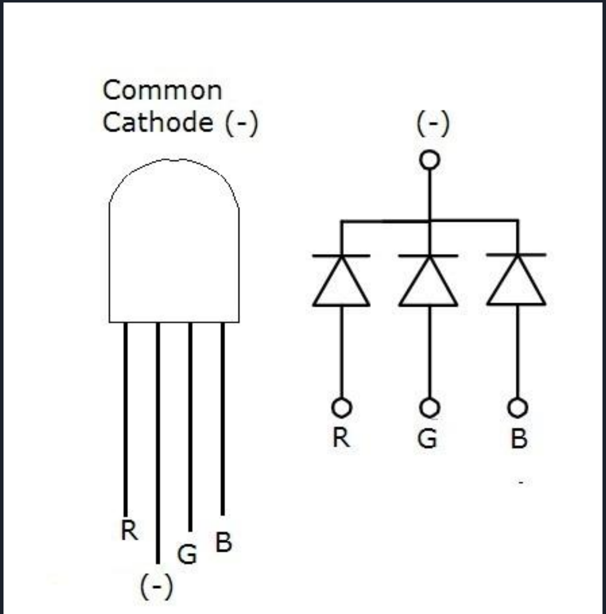
```
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db
import Adafruit_DHT

DHT_SENSOR = Adaruit_DHT.DHT22
DHT_PIN = 4

cred = credentials.Certificate(  “PATH_TO_SERVICE_ACCOUNT_FILE.json”  )
firebase_admin.initialize_app( cred, { “databaseURL”:  “LINK_TO_DATABASE_URL” } )
```

Challenge 1 -- Solution Cont.

```
while 1:
        hum, temp = Adaruit_DHT.read_retry(  DHT_SENSOR,  DHT_PIN  )

        temp_ref = db.reference(  “Sensors/temperature”  )
        temp_ref.set( temp )

        hum_ref = db.reference(  “Sensors/humidity”  )
        hum_ref.set( hum )


        print( “Humidity =”, hum )
        print( “Temperature =”, temp)

        time.sleep( 10 )
```

Challenge 2 -- Revisiting RGB LED

Write a python script that read the state of the RGB led and take appropriate action based on the state of the LED.



Challenge 2 -- Solution

```
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db
import RPi.GPIO as GPIO

GPIO.setmode( GPIO.BCM  )
GPIO.setwarnings( False )

RED_PIN = 17
GREEN_PIN = 27
BLUE_PIN = 22

GPIO.setup(  RED_PIN,   GPIO.OUT  )
GPIO.setup(  GREEN_PIN,   GPIO.OUT  )
GPIO.setup(  BLUE_PIN,   GPIO.OUT  )
```

```
### Validate Credentials and Initialize app
cred = credentials.Certificate(
        “PATH_TO_SERVICE_ACCOUNT_FILE.json”  )

firebase_admin.initialize_app( cred,
        { “databaseURL”:  “LINK_TO_DATABASE_URL” } )
```

Challenge 2 -- Solution Cont.

```
def initialize_led(  led_ref  ):
        data = led_ref.get( )

        if data[ “red” ] == True:
                GPIO.output( RED_PIN, GPIO.HIGH )
        else:
                GPIO.output( RED_PIN, GPIO.LOW )

        ### Similar login for Green and Blue LED
```

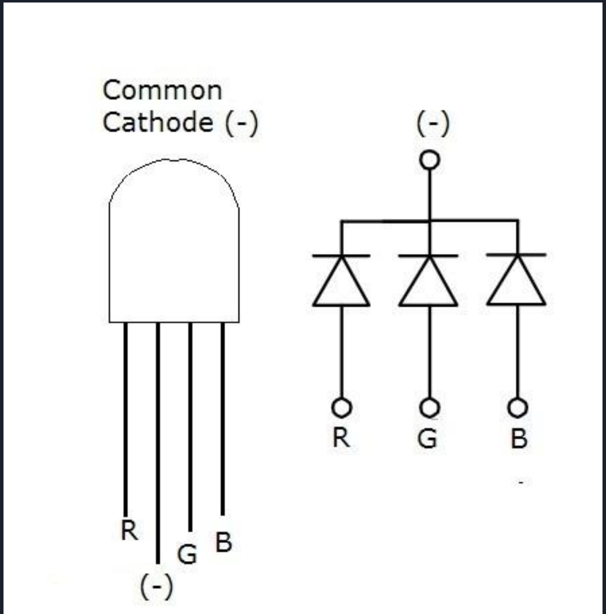
```
## Call the initializer
led_ref = db.reference(  “LEDs/LED1”  )
initialize_led(  led_ref  )
```

4. Event Listener

```
def callback_fun( event ):

        ## do something on callback

db_ref = db.reference( “PATH_TO_REFERENCE” )

db_ref.listen( callback_fun )
```

Challenge 3 -- RGB LED Event Listener

Write a python script that listens to a LED reference on the Firebase Database and changes color based on the Truth value of RGB component in the Database.



Challenge 3 -- Solution

```
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db
import RPi.GPIO as GPIO

GPIO.setmode( GPIO.BCM  )
GPIO.setwarnings( False )

RED_PIN = 17
GREEN_PIN = 27
BLUE_PIN = 22

GPIO.setup(  RED_PIN,   GPIO.OUT  )
GPIO.setup(  GREEN_PIN,   GPIO.OUT  )
GPIO.setup(  BLUE_PIN,   GPIO.OUT  )
```

```
def led_callback(  event  ):
        data = event.data
        path = event.path

        if “red” in path and type( data ) is bool:
                GPIO.output( RED_PIN, GPIO.HIGH )
        else:
                GPIO.output( RED_PIN, GPIO.LOW )

        ### Similar login for Green and Blue LED
```

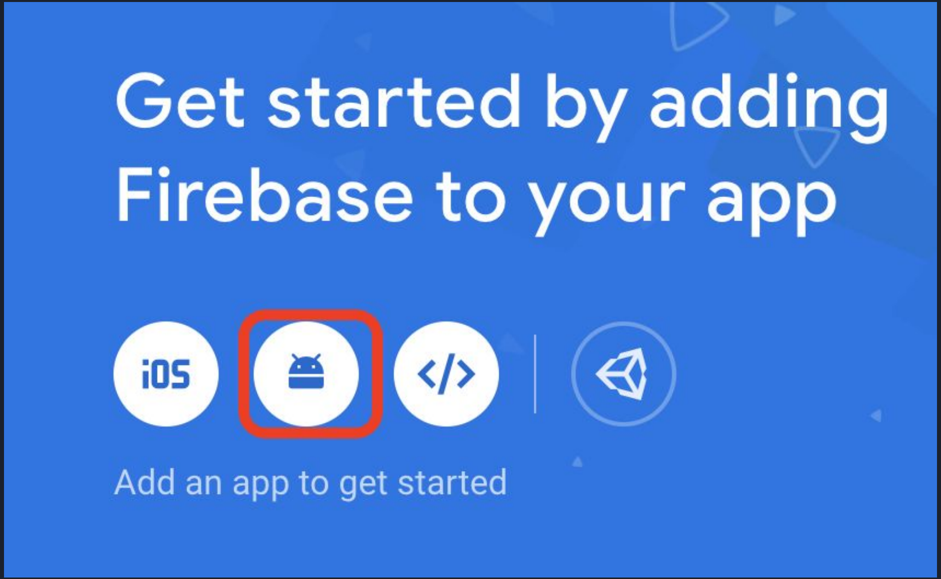
Challenge 3 -- Solution Cont.

```
### Validate Credentials and Initialize app
cred = credentials.Certificate(  “PATH_TO_SERVICE_ACCOUNT_FILE.json”  )
firebase_admin.initialize_app( cred, { “databaseURL”:  “LINK_TO_DATABASE_URL” } )

### Set the Event Listener
led_ref = db.reference(   “LEDs/LED1”   )
initialize_led(   led_ref   )
led_ref.listen(   led_callback   )
```

Firebase with Android Studio

Refer to the following link for the official documentation

1. Registering Your Application

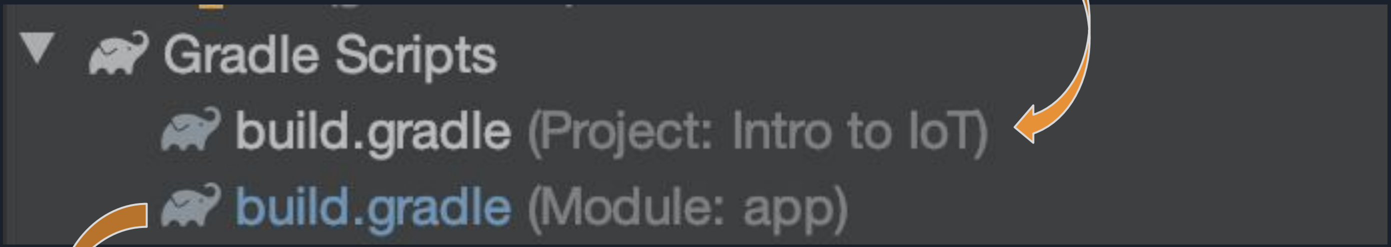


Get started by adding Firebase to your app

Start by clicking Android Icon in the Project Overview Page.

Locating Project vs App build.gradle file

Project level build.gradle file



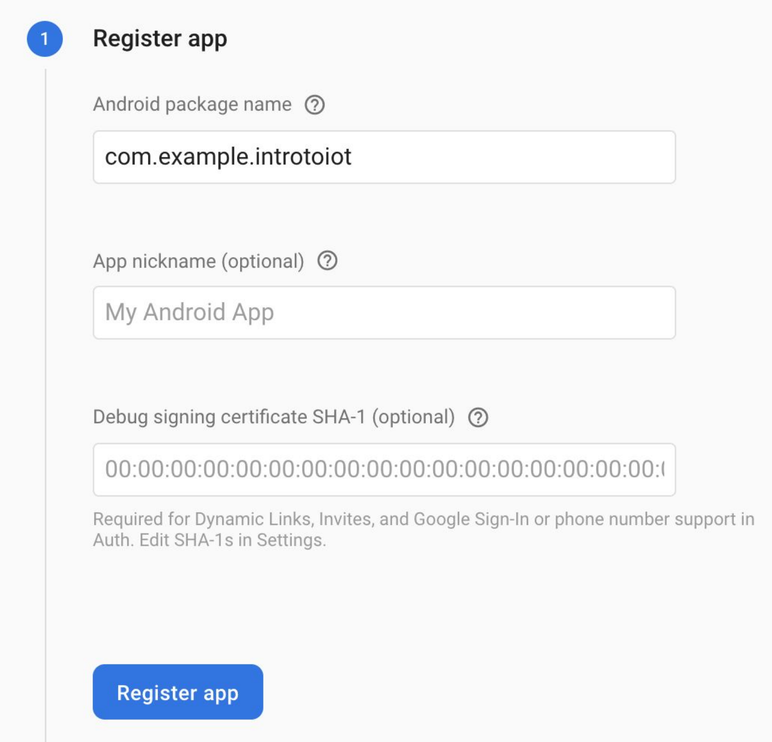
▼ Gradle Scripts

build.gradle (Project: Intro to IoT)

build.gradle (Module: app)

App level build.gradle file

2. Adding Package Name



1 Register app

Android package name ⓘ

com.example.introtoiot

App nickname (optional) ⓘ

My Android App

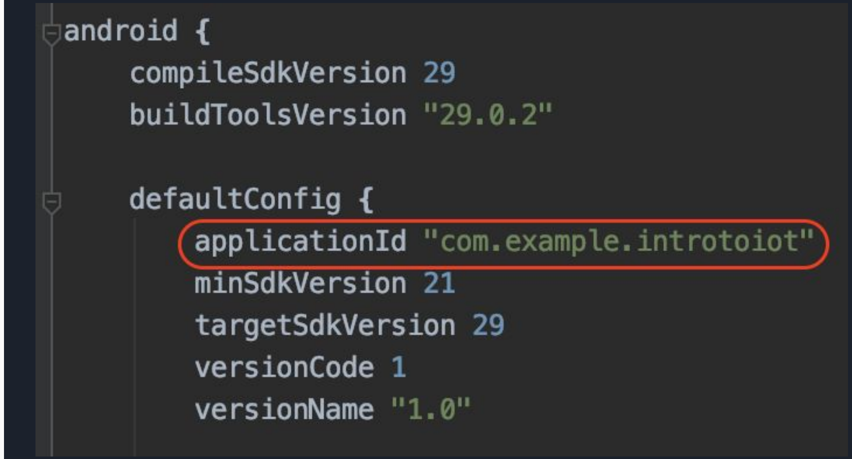
Debug signing certificate SHA-1 (optional) ⓘ

00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:(

Required for Dynamic Links, Invites, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.

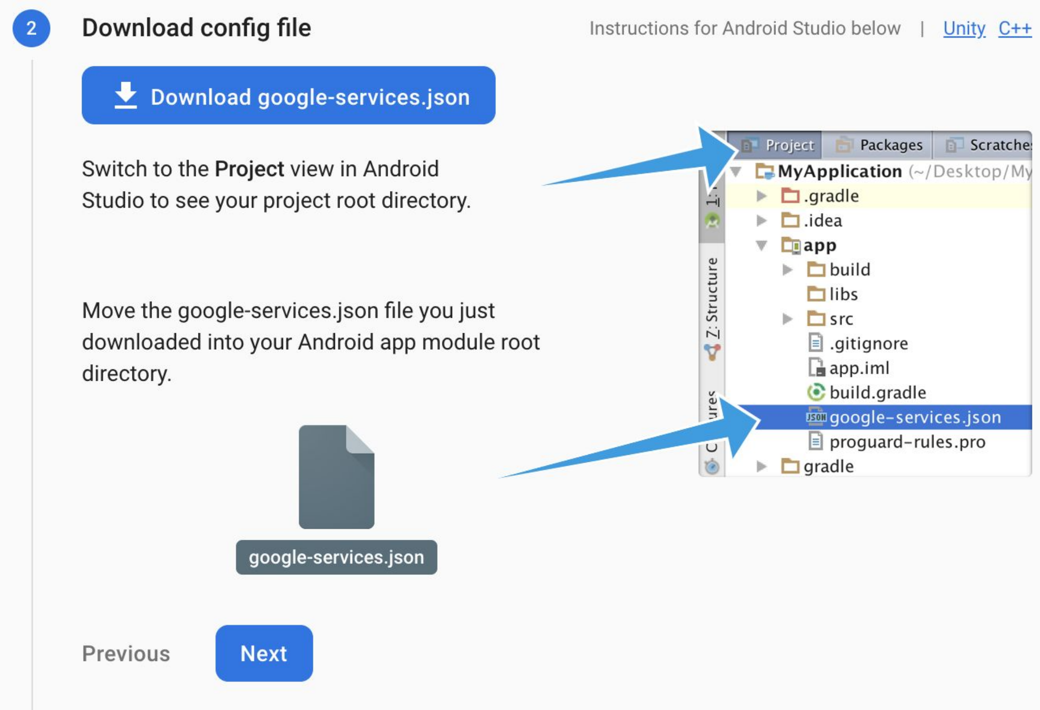
Register app

For the Android package name, look for applicationId in the App level build.gradle file



```
android {
    compileSdkVersion 29
    buildToolsVersion "29.0.2"

    defaultConfig {
        applicationId "com.example.introtoiot"
        minSdkVersion 21
        targetSdkVersion 29
        versionCode 1
        versionName "1.0"
```

3. Downloading the Service File



2 Download config file

Instructions for Android Studio below | Unity C++

Download google-services.json

Switch to the **Project** view in Android Studio to see your project root directory.

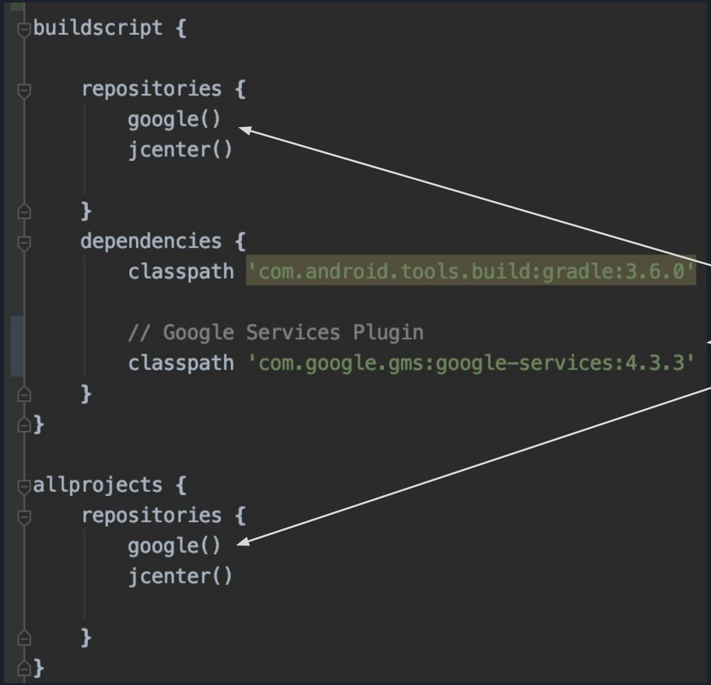
Move the google-services.json file you just downloaded into your Android app module root directory.

Previous

Next

Download the google-services.json file and save it in under the app folder of the project.

4.1. Project level build.gradle file

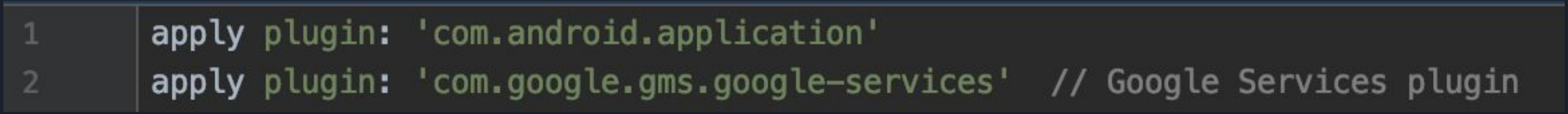


```
buildscript {

    repositories {
        google()
        jcenter()

    }
    dependencies {
        classpath 'com.android.tools.build:gradle:3.6.0'

        // Google Services Plugin
        classpath 'com.google.gms:google-services:4.3.3'
    }
}

allprojects {
    repositories {
        google()
        jcenter()

    }
}
```

Verify the following three in Project level build.gradle file.

If it's not there, add it

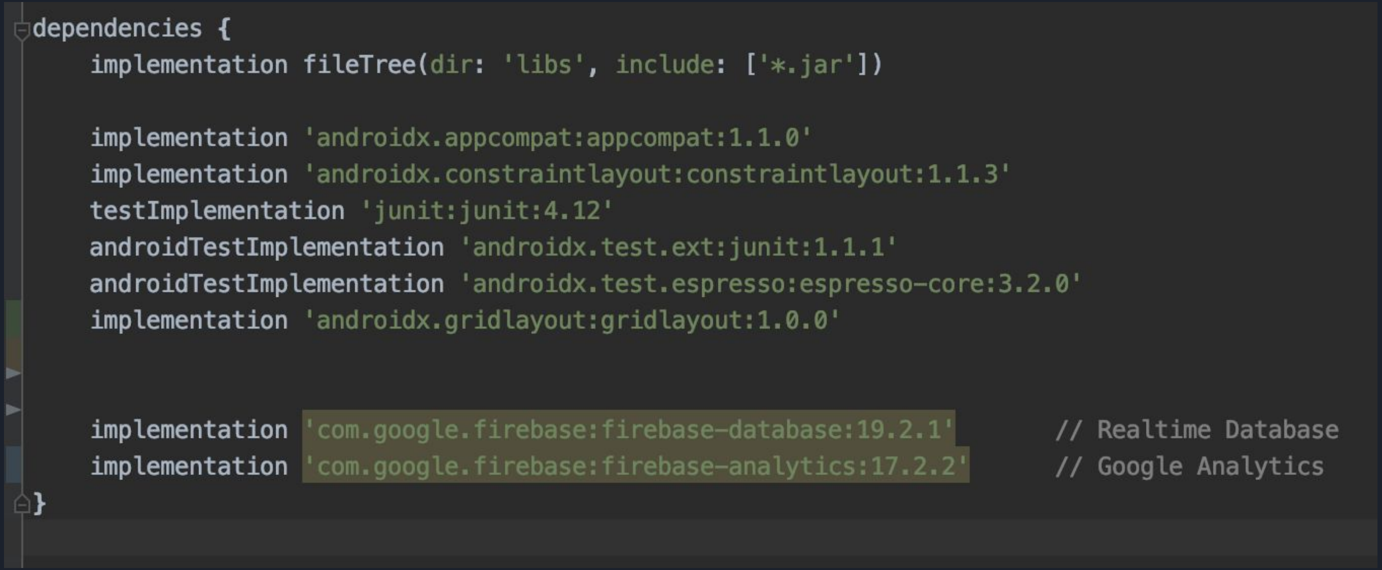
4.2. App level build.gradle file



```
apply plugin: 'com.android.application'
apply plugin: 'com.google.gms.google-services'  // Google Services plugin
```

Add the Google Service Plugin on top
of the App level build.gradle file.

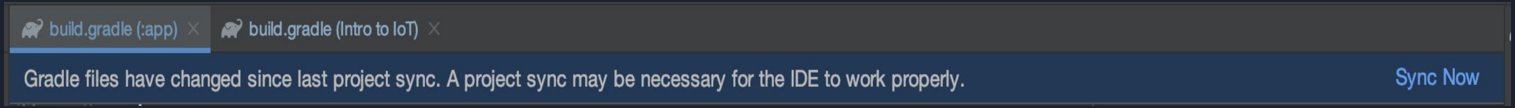
4.3. App level build.gradle file



```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])

    implementation 'androidx.appcompat:appcompat:1.1.0'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test.ext:junit:1.1.1'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'
    implementation 'androidx.gridlayout:gridlayout:1.0.0'


    implementation 'com.google.firebase:firebase-database:19.2.1'       // Realtime Database
    implementation 'com.google.firebase:firebase-analytics:17.2.2'      // Google Analytics
}
```

Add at least the Realtime-Database dependencies in the App level build.gradle file.

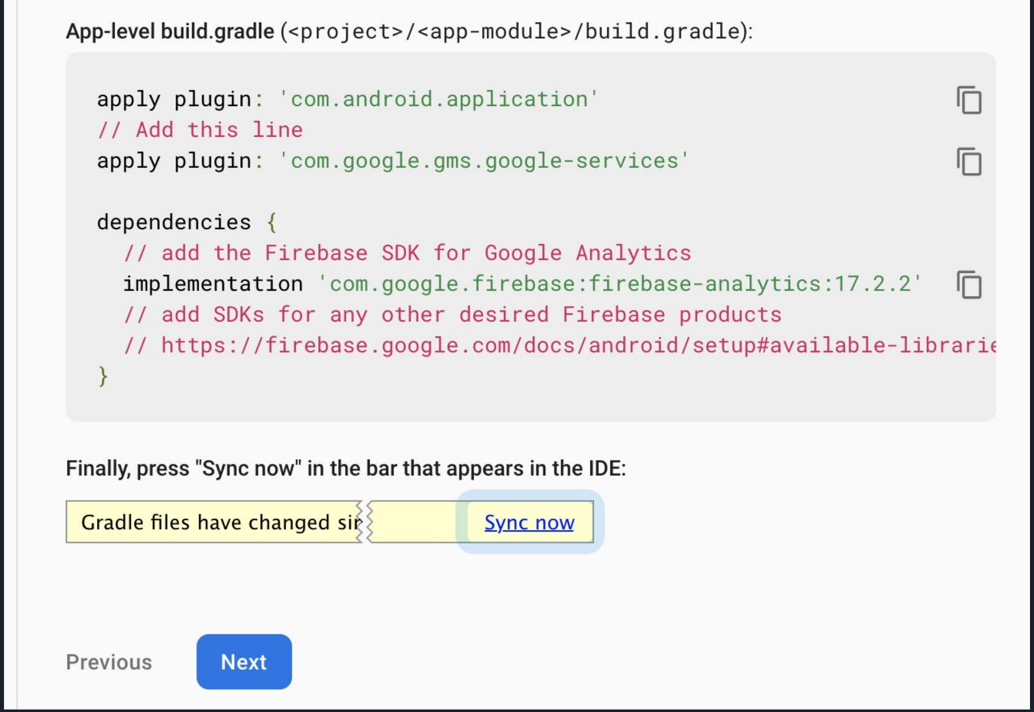
4.4. Sync Gradle Files



Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly.

Sync the Gradle file with the Sync Now button that appears at the top of the IDE.

4.5. Finish Adding Firebase SDK



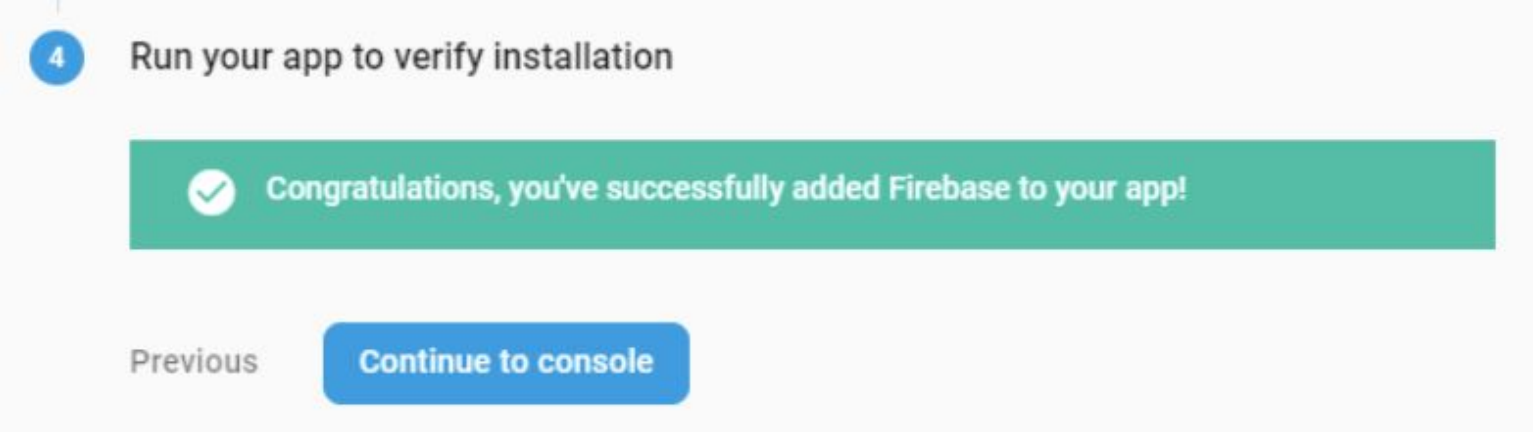
App-level build.gradle (`<project>/<app-module>/build.gradle`):

```
apply plugin: 'com.android.application'
// Add this line
apply plugin: 'com.google.gms.google-services'

dependencies {
  // add the Firebase SDK for Google Analytics
  implementation 'com.google.firebase:firebase-analytics:17.2.2'
  // add SDKs for any other desired Firebase products
  // https://firebase.google.com/docs/android/setup#available-librarie
}
```

Finally, press "Sync now" in the bar that appears in the IDE:

Click Next on the Firebase setup page.

5. Run App for Verification



④ Run your app to verify installation

✓ Congratulations, you've successfully added Firebase to your app!

Make sure Firebase is successfully added
to the app before continuing to console

Using Firebase with Android Studio

Writing to Firebase Database

```
DatabaseReference myDatabase = FirebaseDatabase.getInstance( ).getReference( );
```

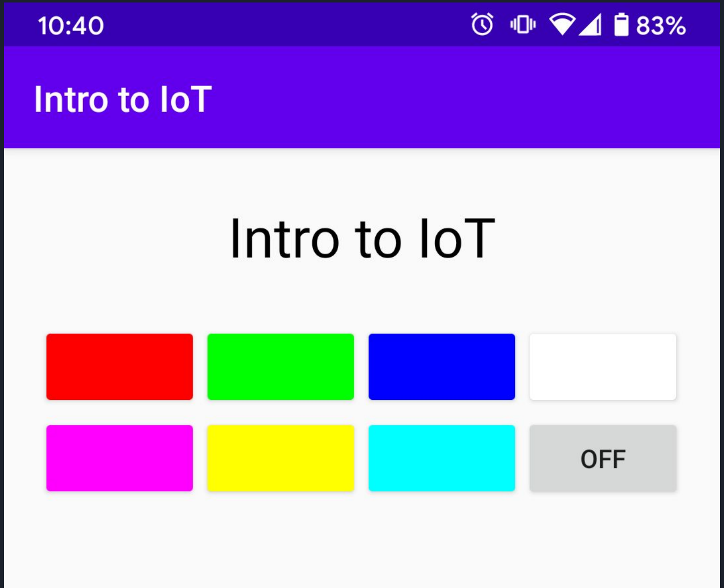
First: Create a database reference

```
myDatabase.child( “Users” ).child( “ECE 30” ).child( “email” ).setValue( “ece30@gmail.com” );
```

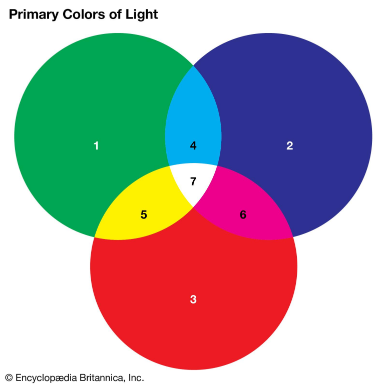
Then traverse through the tree to set the appropriate value

Link to the official documentation

Challenge 1: Saving Color Values

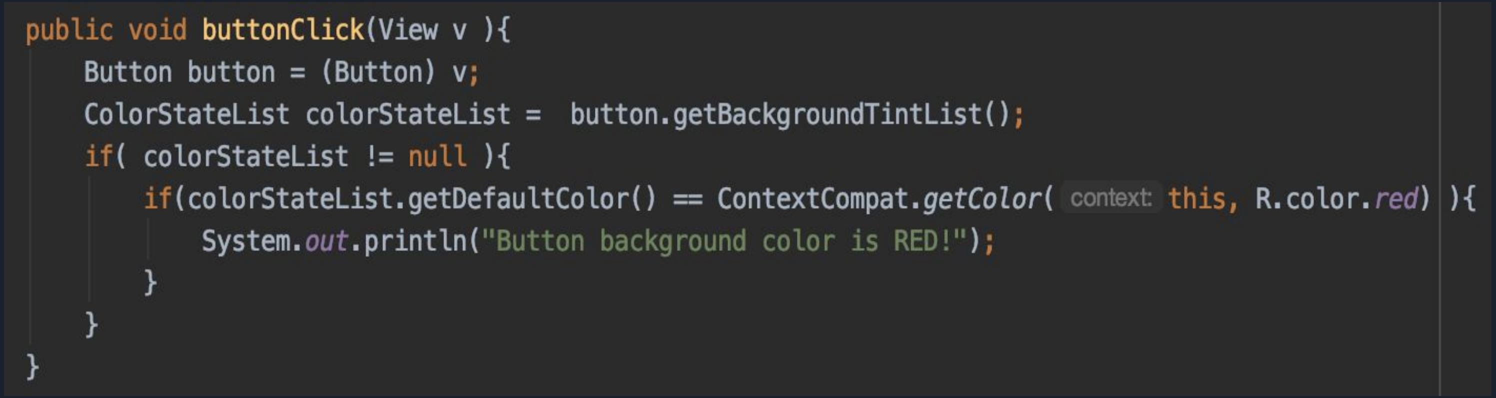


- Modify the onClick listener for the color buttons to save the appropriate color values for the button click
- Modify the existing color values for the choice of LED



© Encyclopædia Britannica, Inc.

Button Background Color



```
public void buttonClick(View v ){
    Button button = (Button) v;
    ColorStateList colorStateList =  button.getBackgroundTintList();
    if( colorStateList != null ){
        if(colorStateList.getDefaultColor() == ContextCompat.getColor( context: this, R.color.red) ){
            System.out.println("Button background color is RED!");
        }
    }
}
```

Use the following template to retrieve the background color of any View

Reading From Firebase Database

```
DatabaseReference myDatabase = FirebaseDatabase.getInstance( ).getReference( “path/to/ref” );
```

First: Create a database reference

```
myDatabase.addListenerForSingleValueEvent( new ValueEventListener( ) { …. }  );
```

Then add a single value event listener to send a read request to the database

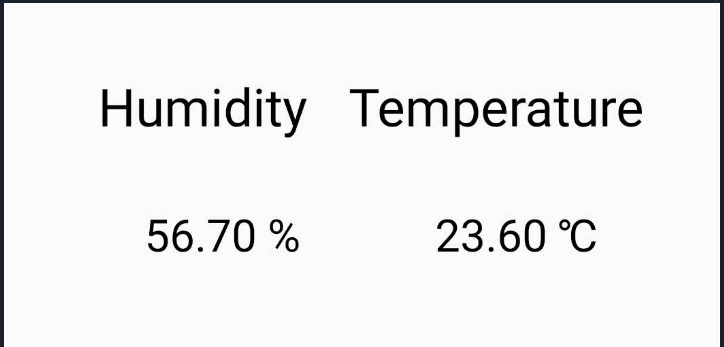
Link to the official documentation

DataSnapshot

```
@Override

public void onDataChange( @NonNull Datasnapshot dataSnapshot ) {

        String data = (String) dataSnapshot.getValue( );

        System.out.println( “Data = “ + data );

}
```

Link to the official documentation

Challenge: 2 Sensor TextField



Humidity	Temperature
56.70 %	23.60 ℃

- Modify the code so that humidity value gets updated as soon as the app launches
- Modify the code so that temperature value gets updated as soon as the app launches

Reading From Firebase Database

```
DatabaseReference myDatabase = FirebaseDatabase.getInstance( ).getReference( “path/to/ref” );
```

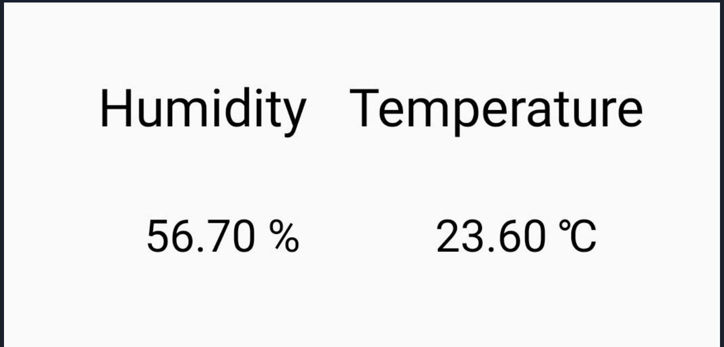
First: Create a database reference

```
myDatabase.addValueEventListener( new ValueEventListener( ) { …. }  );
```

Then add a event listener for consistent updates to the database

Link to the official documentation

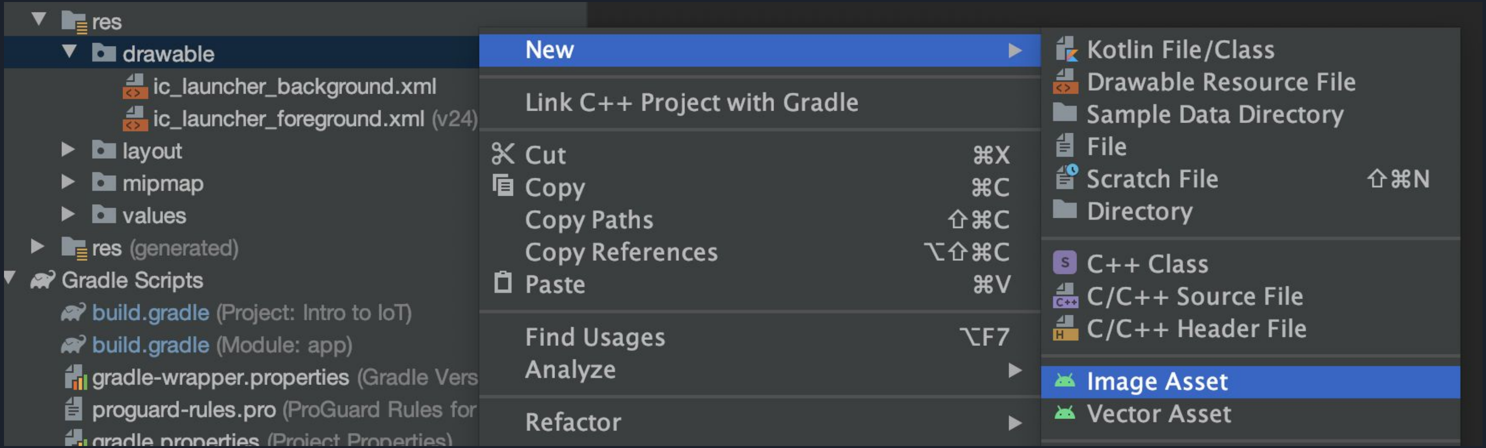
Challenge 3: Sensor TextField Event Listener



Humidity	Temperature
56.70 %	23.60 ℃

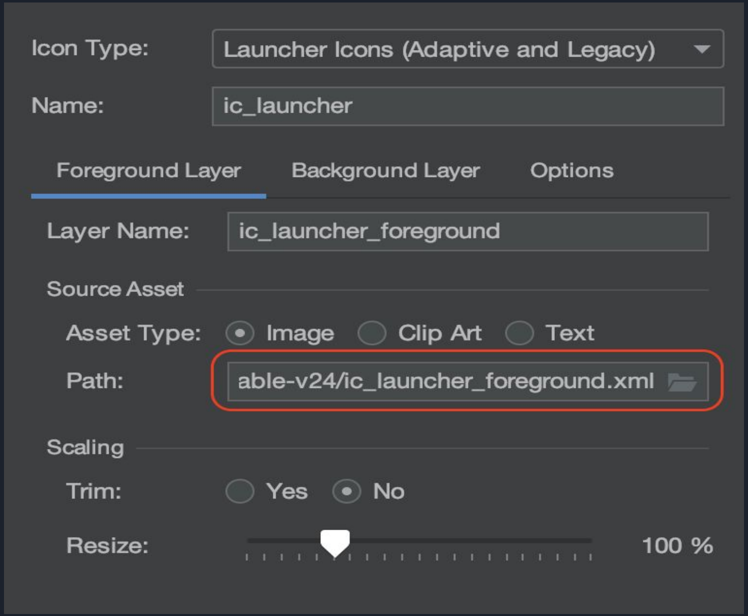
- Modify the code so that humidity value gets updated each time it’s modified in the database
- Modify the code so that temperature value gets updated each time it’s modified in the database

Updating App Icon



Right click on the Drawable folder under the Resources section. Click on New → Image Asset

Select the Icon



Icon Type: Launcher Icons (Adaptive and Legacy)

Name: ic_launcher

Layer Name: ic_launcher_foreground

Source Asset

Asset Type: ● Image ○ Clip Art ○ Text

Path: able-v24/ic_launcher_foreground.xml

Scaling

Trim: ○ Yes ● No

Resize: 100 %

Select the Icon from your local machine

Customize the Foreground Layer

Customize the Background Layer

Customize anything else to your liking

Click Next → Click Finish

Link to a possible safe image

Questions?