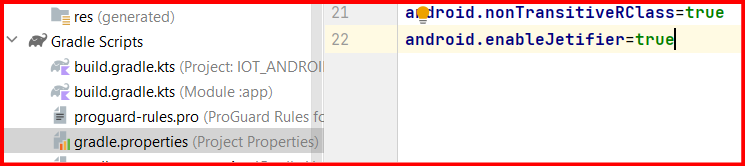
Grandle( Mobile App )

*dependencies* **{** *implementation*(**"androidx.appcompat:appcompat:1.6.1"**)  
 *implementation*(**"com.google.android.material:material:1.5.0"**)  
 *implementation*(**"androidx.constraintlayout:constraintlayout:2.1.4"**)  
 *testImplementation*(**"junit:junit:4.13.2"**)  
 *androidTestImplementation*(**"androidx.test.ext:junit:1.1.15"**)  
*// androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")  
 androidTestImplementation*(**"androidx.test.espresso:espresso-core:3.4.0"**)  
  
  
 *implementation*(**"org.eclipse.paho:org.eclipse.paho.client.mqttv3:1.1.0"**)  
 *implementation*(**"org.eclipse.paho:org.eclipse.paho.android.service:1.1.1"**)  
**}**

Androidmanifest

<**uses-permission android:name="android.permission.WAKE\_LOCK"**/>  
<**uses-permission android:name="android.permission.INTERNET"**/>  
<**uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE"**/>  
<**uses-permission android:name="android.permission.READ\_PHONE\_STATE"** />

<**service android:name="org.eclipse.paho.android.service.MqttService"** />



**android.enableJetifier**=**true**

MainActivity

**package** com.example.mqtt\_appandr\_demo;  
  
  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** android.os.Bundle;  
**import** android.util.Log;  
**import** android.view.View;  
**import** android.widget.ImageView;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** org.eclipse.paho.android.service.MqttAndroidClient;  
**import** org.eclipse.paho.client.mqttv3.IMqttActionListener;  
**import** org.eclipse.paho.client.mqttv3.IMqttDeliveryToken;  
**import** org.eclipse.paho.client.mqttv3.IMqttToken;  
**import** org.eclipse.paho.client.mqttv3.MqttCallback;  
**import** org.eclipse.paho.client.mqttv3.MqttClient;  
**import** org.eclipse.paho.client.mqttv3.MqttConnectOptions;  
**import** org.eclipse.paho.client.mqttv3.MqttException;  
**import** org.eclipse.paho.client.mqttv3.MqttMessage;  
  
**import** java.io.UnsupportedEncodingException;  
**import** java.nio.charset.StandardCharsets;  
  
  
**public class** MainActivity **extends** AppCompatActivity {  
 **private** ImageView **den**;  
 **private** TextView **Temp**;  
 **private int c1**=1;  
 MqttAndroidClient **mqttAndroidClient**;  
 **private static final** String ***Tag*** = **""**;  
 String **clientID** = MqttClient.*generateClientId*();  
 MqttAndroidClient **client**;  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 **den** = findViewById(R.id.*imgden*);  
 **Temp** = findViewById(R.id.*txtTemp*);  
 MQTT();  
 }  
  
 **public void** MQTT() {  
 String clientID = MqttClient.*generateClientId*();  
 **final** MqttAndroidClient client = **new** MqttAndroidClient(**this**.getApplicationContext(), **"tcp://broker.hivemq.com:1883"**, clientID);  
 MqttConnectOptions options = **new** MqttConnectOptions();  
 options.setMqttVersion(MqttConnectOptions.***MQTT\_VERSION\_3\_1***);  
 options.setCleanSession(**false**);  
  
 *// Remove these lines if not required  
 // options.setUserName("");  
 // options.setPassword("".toCharArray());* **try** {  
 **final** IMqttToken token = client.connect(options);  
 token.setActionCallback(**new** IMqttActionListener() {  
 @Override  
 **public void** onSuccess(IMqttToken asyncActionToken) {  
 Toast.*makeText*(MainActivity.**this**, **"Connected MQTT server"**, Toast.***LENGTH\_SHORT***).show();  
 SUB(client, **"den"**);  
 SUB(client, **"Temp"**);  
 client.setCallback(**new** MqttCallback() {  
 @Override  
 **public void** connectionLost(Throwable cause) {  
 Toast.*makeText*(MainActivity.**this**, **"Mat ket noi Server"**, Toast.***LENGTH\_SHORT***).show();  
 }  
  
 @Override  
 **public void** messageArrived(String topic, MqttMessage message) **throws** Exception {  
 *// control den* **if** (topic.equals(**"den"**)) {  
 **if** (message.toString().equals(**"1"**)) {  
 **den**.setBackgroundResource(R.drawable.*light\_on*);  
 **c1** = 0;  
 }  
 **if** (message.toString().equals(**"0"**)) {  
 **den**.setBackgroundResource(R.drawable.*light\_off*);  
 **c1** = 1;  
 }  
 }  
 *// lay du lieu nhiet do tu MQTT ve* **if**(topic.equals(**"Temp"**))  
 {  
 **Temp**.setText(message.toString()+**"℃"**);  
 }  
 }  
  
 @Override  
 **public void** deliveryComplete(IMqttDeliveryToken token) {  
 }  
 });  
  
 **den**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
*// Toast.makeText(MainActivity.this, "den", Toast.LENGTH\_SHORT).show();* **c1**++;  
 **if**(**c1**%2==0)  
 {  
 String topic = **"den"**;  
 String payload = **"1"**;  
 **byte**[] encodePayload = **new byte**[0];  
 **try** {  
 encodePayload = payload.getBytes(**"UTF-8"**);  
 MqttMessage message = **new** MqttMessage(encodePayload);  
 message.setRetained(**true**);  
 client.publish(topic, message);  
 }  
 **catch** (UnsupportedEncodingException | MqttException e)  
 {  
 e.printStackTrace();  
 }  
 }  
 **else** {  
 String topic = **"den"**;  
 String payload = **"0"**;  
 **byte**[] encodePayload = **new byte**[0];  
 **try** {  
 encodePayload = payload.getBytes(**"UTF-8"**);  
 MqttMessage message = **new** MqttMessage(encodePayload);  
 message.setRetained(**true**);  
 client.publish(topic, message);  
 }  
 **catch** (UnsupportedEncodingException | MqttException e)  
 {  
 e.printStackTrace();  
 }  
 }  
 }  
 });  
 }  
  
 @Override  
 **public void** onFailure(IMqttToken asyncActionToken, Throwable exception) {  
 Toast.*makeText*(MainActivity.**this**, **"that bai"**, Toast.***LENGTH\_SHORT***).show();  
 Log.*d*(***Tag***, **"onFailure"**);  
 }  
 });  
 } **catch** (MqttException e) {  
 e.printStackTrace();  
 }  
 } *// end MQTT* **public void** SUB( MqttAndroidClient client, String topic)  
 {  
 **int** qos = 1;  
 **try** {  
 IMqttToken subToken = client.subscribe(topic, qos);  
 subToken.setActionCallback(**new** IMqttActionListener() {  
 @Override  
 **public void** onSuccess(IMqttToken asyncActionToken) {  
 *// the message was published* }  
  
 @Override  
 **public void** onFailure(IMqttToken asyncActionToken, Throwable exception) {  
  
 }  
 });  
  
 }  
 **catch** (MqttException e)  
 {  
 e.printStackTrace();  
 }  
 }*//end SUB*}