

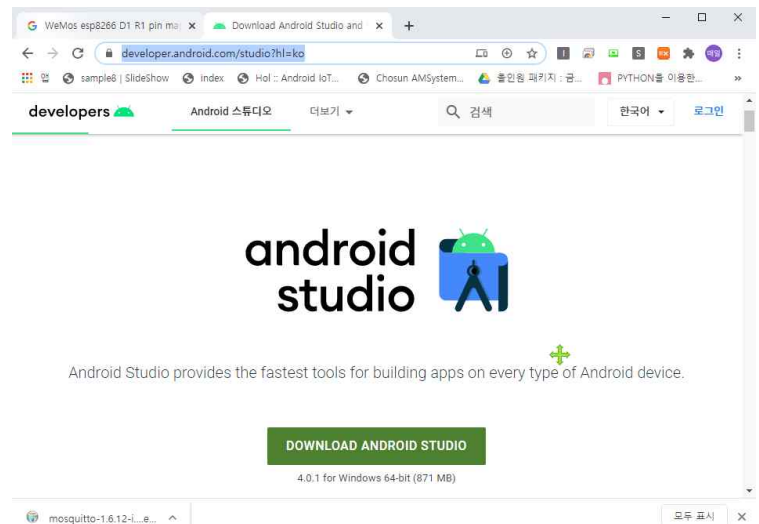
8. Android에서 제어하기

8.1 개발환경 구성

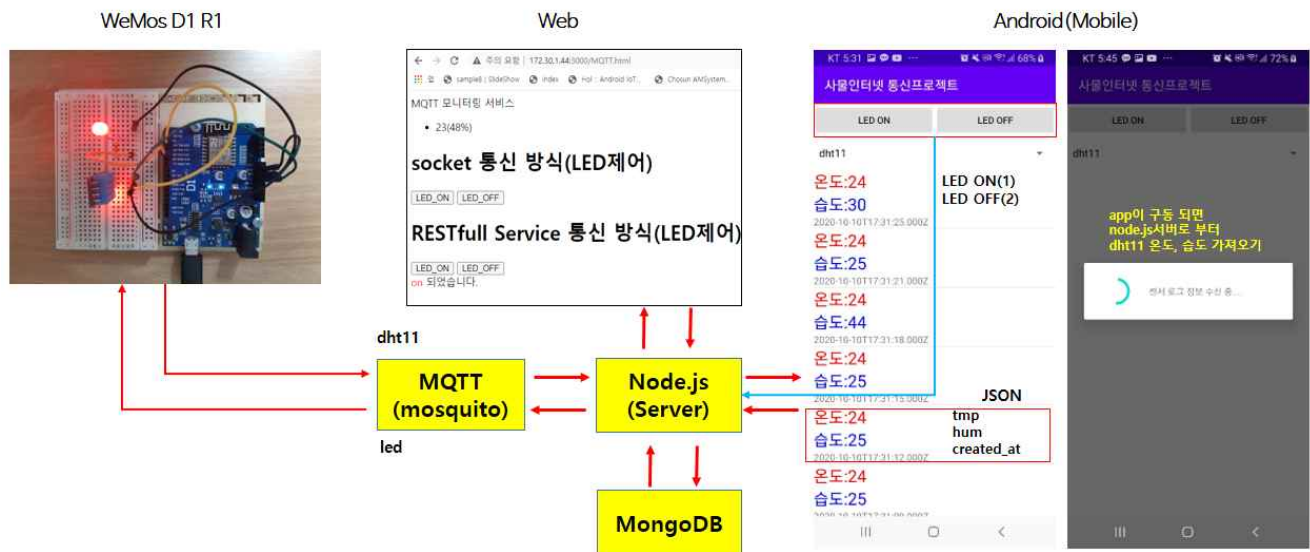
* android studio 설치하기

아래사이트에서 안드로이드 스튜디오를 다운받아 설치한다.

<https://developer.android.com/studio?hl=ko>



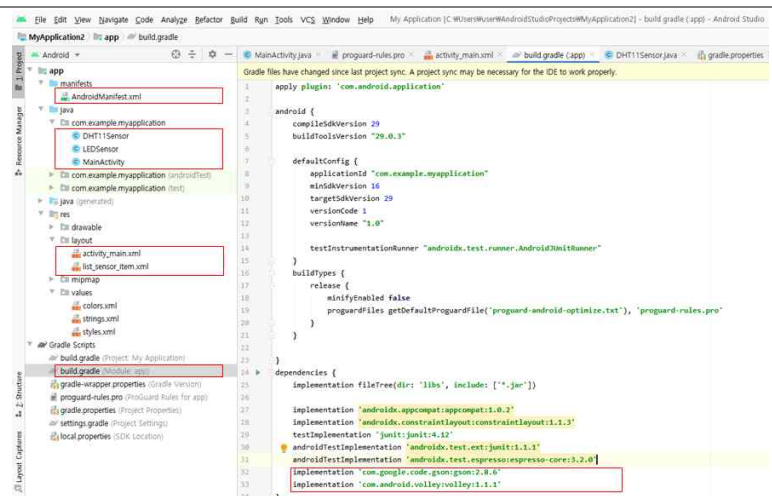
[안드로이드에서 구동화면]



프로젝트 만들기

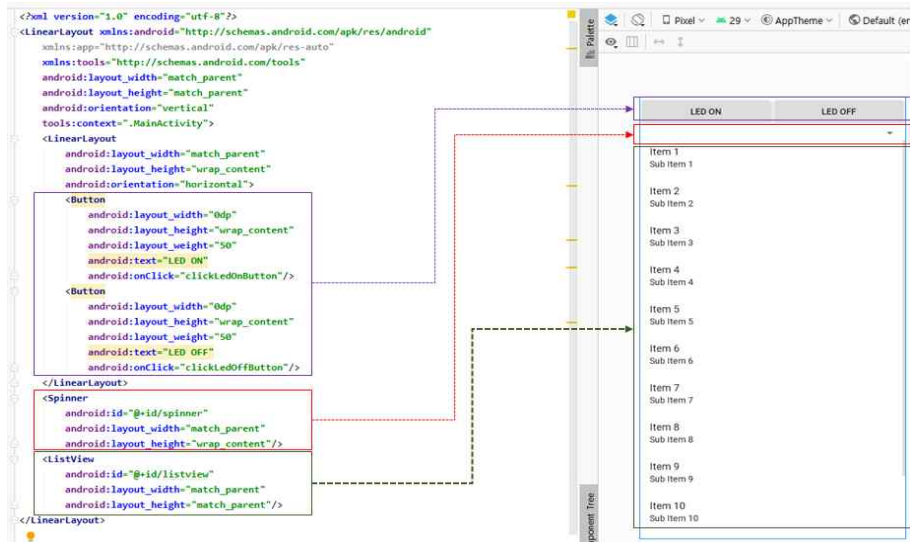
- 아래 6개의 파일을 만들거나 수정한다.

1. AndroidManifest.xml(수정)
2. MainActivity.java(수정)
3. DHT11Sensor.java(만들기)
4. LEDSensor.java(만들기)
5. activity_main.xml(수정)
6. list_sensor_item.xml(만들기)
7. build.gradle(수정)



8.2 레이아웃 구성하기

activity_main.xml



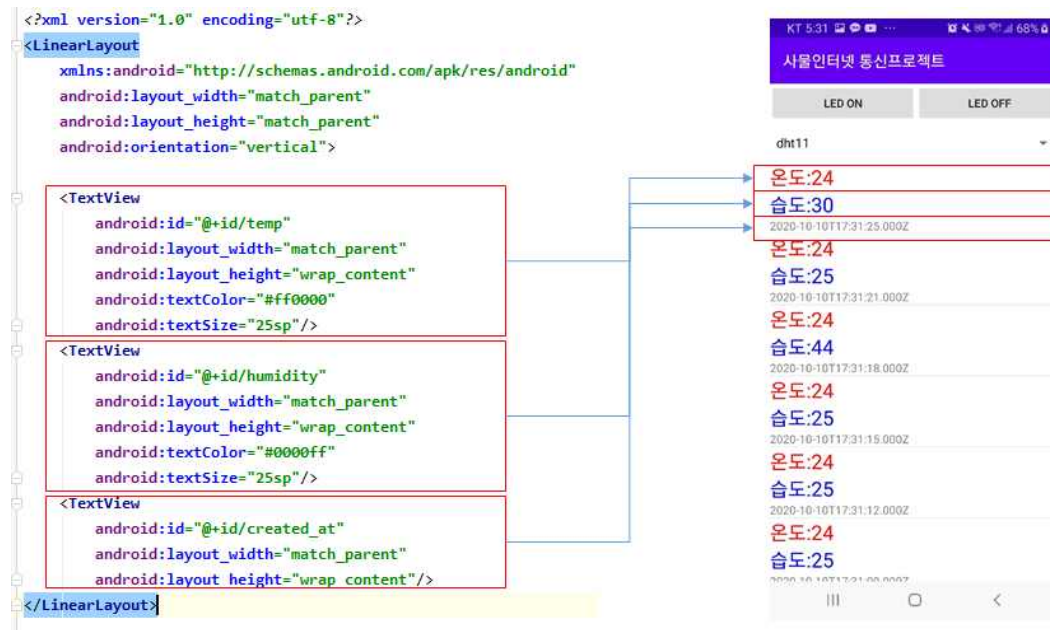
```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <Button
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="50"
            android:text="LED ON"
            android:onClick="clickLedOnButton"/>
        <Button
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="50"
            android:text="LED OFF"
            android:onClick="clickLedOffButton"/>
    </LinearLayout>
    <Spinner
        android:id="@+id/spinner"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
    <ListView
        android:id="@+id/listview"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>
</LinearLayout>

```

8.2 레이아웃 구성하기

list_sensor_item.xml



```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout
```

```
    xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
```

```
    android:orientation="vertical")
```

```
<TextView
```

```
    android:id="@+id/temp"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    android:textColor="#ff0000"
```

```
    android:textSize="25sp"/>
```

```
<TextView
```

```
    android:id="@+id/humidity"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    android:textColor="#0000ff"
```

```
    android:textSize="25sp"/>
```

```
<TextView
```

```
    android:id="@+id/created_at"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"/>
```

```
</LinearLayout>
```

8.3 메인 액티비티 만들기

MainActivity.java

```

package com.example.myapplication;

import android.app.ProgressDialog;
import android.content.Context;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import org.json.JSONArray;
import org.json.JSONObject;
import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // 여기서부터 코딩 시작~~
        final String[] sensors={"dht11","mq2"};
        ArrayAdapter<String> spinnerAdapter=
            new ArrayAdapter<String>(MainActivity.this,
                android.R.layout.simple_spinner_dropdown_item, sensors);
        // Alt+Enter
        Spinner spinner=(Spinner)findViewById(R.id.spinner);
        spinner.setAdapter(spinnerAdapter);
        //스피너 선택시 동작되는 부분 : 센서 데이터 가져오기
        spinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {
                final ProgressDialog dialog=new ProgressDialog(MainActivity.this);
                dialog.setMessage("센서 로그 정보 수신 중....");
                dialog.show();
            }
        });
    }
}

```

```

        Response.Listener<String> listener=new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                dialog.dismiss();
                try {
                    JSONArray array=new JSONArray(response);
                    items.clear();
                    for(int i=0; i<array.length();i++){
                        JSONObject obj=array.getJSONObject(i);
                        items.add(new Item(obj.getInt("tmp"),
                            obj.getInt("hum"),
                            obj.getString("created_at")));
                    }
                    ItemAdapter adapter=new ItemAdapter(MainActivity.this);
                    ListView listView=(ListView)findViewById(R.id.listview);
                    listView.setAdapter(adapter);
                }catch (Exception e){
                    e.printStackTrace();
                }
            }
        };

        StringRequest dht11=new DHT11Sensor(sensors[i], listener);
        dht11.setShouldCache(false);
        RequestQueue requestQueue = Volley.newRequestQueue(MainActivity.this);
        requestQueue.add(dht11);

    }
    @Override
    public void onNothingSelected(AdapterView<?> adapterView) {
    }
    });
}

class Item{
    int temp, humidity; String created_at;
    Item(int temp, int humidity, String created_at){
        this.temp=temp;
        this.humidity=humidity;
        this.created_at=created_at;
    }
}

ArrayList<Item> items=new ArrayList<Item>();
class ItemAdapter extends ArrayAdapter{
    public ItemAdapter(Context context) {

```

```

        super(context, R.layout.list_sensor_item, items);
    }
    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        View view=convertView;
        if(view==null){
            LayoutInflater inflater=
                (LayoutInflater) getSystemService(LAYOUT_INFLATER_SERVICE);
            view=inflater.inflate(R.layout.list_sensor_item, null);
        }
        TextView tempText=view.findViewById(R.id.temp);
        TextView humidityText=view.findViewById(R.id.humidity);
        TextView createdAtText=view.findViewById(R.id.created_at);
        tempText.setText("온도:"+items.get(position).temp);
        humidityText.setText("습도:"+items.get(position).humidity);
        createdAtText.setText("수집정보(날짜/시간>"+items.get(position).created_at);
        return view;
    }
}

// LED ON 버튼 클릭시 동작되는 부분
public void clickLedOnButton(View view) {
    // 서버 응답시 실행되는 콜백 리스너
    Response.Listener<String> listener=new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            try {
                JSONObject obj=new JSONObject(response);
                Toast.makeText(getApplicationContext(),
                    "LED가 켜짐:"+obj.get("led"), Toast.LENGTH_SHORT).show();
            } catch (Exception e){
                e.printStackTrace();
            }
        }
    };

    StringRequest led=new LEDSensor("on", listener);
    led.setShouldCache(false);
    RequestQueue requestQueue = Volley.newRequestQueue(MainActivity.this);
    requestQueue.add(led);
}

// LED OFF 버튼 클릭시 동작되는 부분
public void clickLedOffButton(View view) {
    // 서버 응답시 실행되는 콜백 리스너
    Response.Listener<String> listener=new Response.Listener<String>() {

```

```

@Override
public void onResponse(String response) {
    try {
        JSONObject obj=new JSONObject(response);
        Toast.makeText(getApplicationContext(),
            "LED가 꺼짐:"+obj.get("led"), Toast.LENGTH_SHORT).show();
    }catch (Exception e){
        e.printStackTrace();
    }
}
};

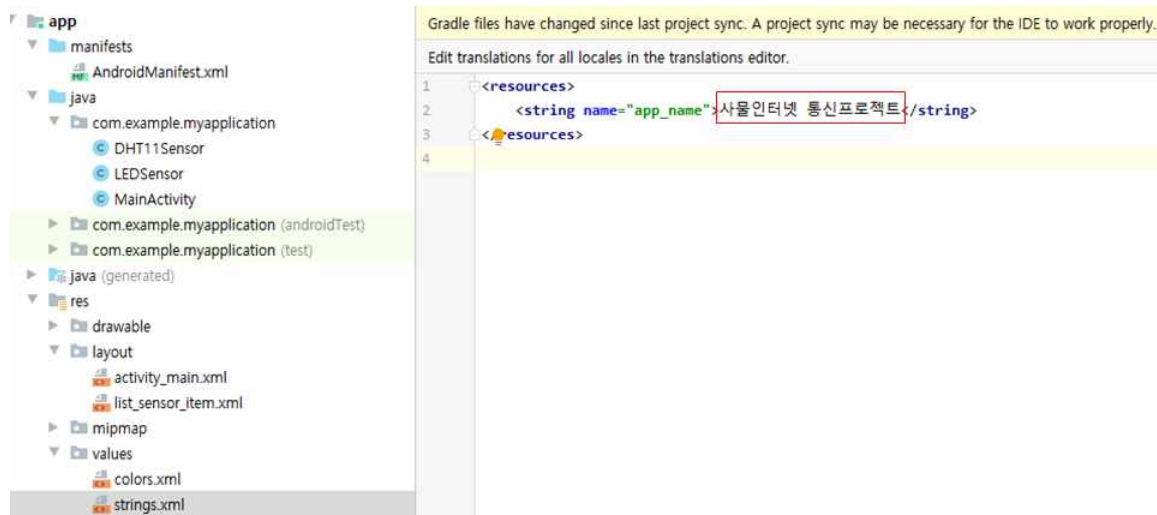
```

```

StringRequest led=new LEDSensor("off", listener);
led.setShouldCache(false);
RequestQueue requestQueue = Volley.newRequestQueue(MainActivity.this);
requestQueue.add(led);
}
}

```

프로젝트 이름 변경하기



8.4 인터넷 및 보안정책 허용	AndroidManifest.xml
<pre> <?xml version="1.0" encoding="utf-8"?> <manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.example.myapplication" <uses-permission android:name="android.permission.INTERNET"/> <application android:usesCleartextTraffic="true" android:allowBackup="true" android:icon="@mipmap/ic_launcher" android:label="@string/app_name" android:roundIcon="@mipmap/ic_launcher_round" android:supportsRtl="true" android:theme="@style/AppTheme"> <activity android:name=".MainActivity"> <intent-filter> <action android:name="android.intent.action.MAIN" /> <category android:name="android.intent.category.LAUNCHER" /> </intent-filter> </activity> </application> </manifest> </pre>	
<p>8.5 volley API 다운로드</p> <p>* build.gradle 파일 하단에 2개의 API를 추가한다.</p> <pre> implementation 'com.google.code.gson:gson:2.8.6' implementation 'com.android.volley:volley:1.1.1' </pre>	<p>build.gradle</p> <pre> dependencies { implementation fileTree(dir: 'libs', include: ['*.jar']) implementation 'androidx.appcompat:appcompat:1.0.2' 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 'com.google.code.gson:gson:2.8.6' implementation 'com.android.volley:volley:1.1.1' } </pre>
<p>8.5.1 DHT11 리스너 만들기</p> <p>* Node.js 서버로 요청하는 주소</p> <p>http://172.30.1.44:3000/devices/device</p>	<p>DHT11Sensor.java</p> <pre> 1 package com.example.myapplication; 2 import com.android.volley.Response; 3 import com.android.volley.toolbox.StringRequest; 4 import java.util.HashMap; 5 import java.util.Map; 6 public class DHT11Sensor extends StringRequest { 7 final static private String URL="http://172.30.1.44:3000/devices/device"; 8 private Map<String, String> parameters; 9 10 public DHT11Sensor(String sensor, Response.Listener<String> listener){ 11 super(Method.POST, URL, listener,null); 12 parameters=new HashMap<>(); 13 parameters.put("sensor",sensor); 14 } 15 16 @Override 17 protected Map<String, String> getParams() { 18 return parameters; 19 } 20 } </pre>

8.5.2 LED 리스너 만들기

LEDSensor.java

* Node.js 서버로 요청하는 주소
<http://172.30.1.44:3000/devices/led>

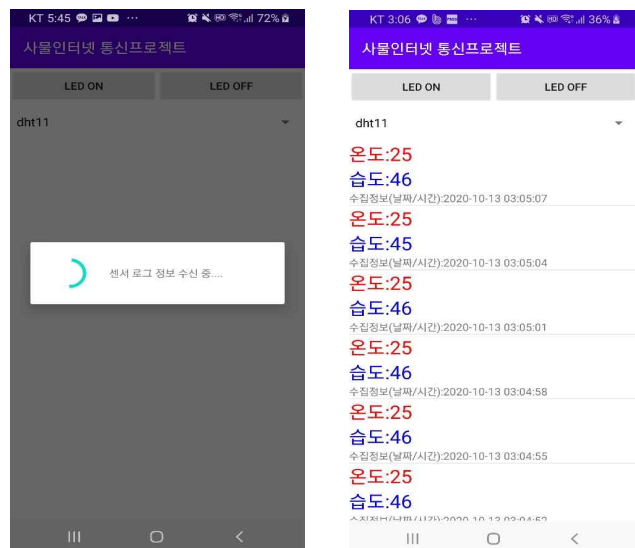
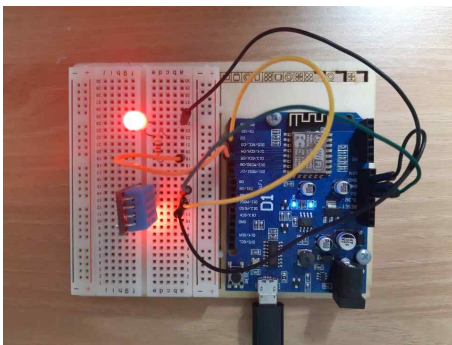
```

1 package com.example.myapplication;
2 import com.android.volley.Response;
3 import com.android.volley.toolbox.StringRequest;
4 import java.util.HashMap;
5 import java.util.Map;
6 public class LEDSensor extends StringRequest {
7     final static private String URL="http://172.30.1.44:3000/devices/led";
8     private Map<String, String> parameters;
9
10    public LEDSensor(String led, Response.Listener<String> listener){
11        super(Method.POST, URL, listener,null);
12        parameters=new HashMap<>();
13        parameters.put("flag",led);
14    }
15
16    @Override
17    protected Map<String, String> getParams() {
18        return parameters;
19    }
20 }

```

[실행결과]

안드로이드와 웹 구동 화면



MQTT 모니터링 서비스

- 24(48%)

socket 통신 방식(LED제어)

LED_ON LED_OFF

RESTfull Service 통신 방식(LED제어)

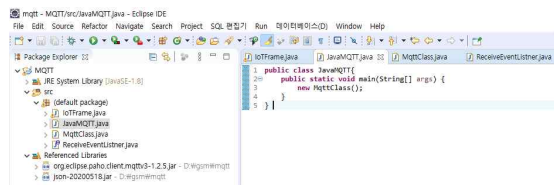
LED_ON LED_OFF

off 되었습니다.

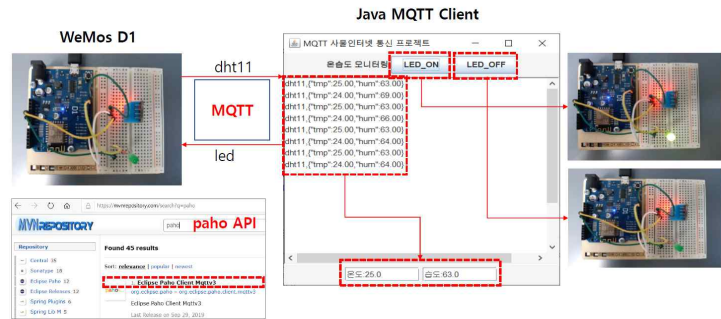
9. Java에서 DHT11 센서 모니터링 및 LED 제어하기(mosquitto 연동)

9.1 개발환경 구성

* eclipse IDE에서 MQTT Java project만들기



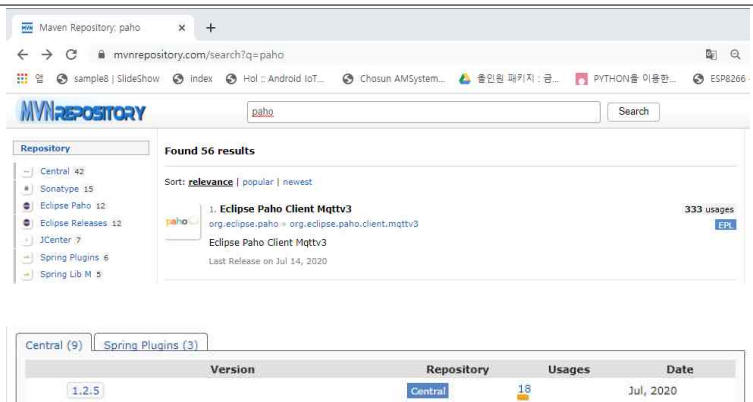
* paho, JSON-Java API를 다운받고 라이브러리 연결하기



9.2 API 다운로드

* paho API 다운로드

아래 사이트에서 paho API를 다운로드한다.
<https://mvnrepository.com/>



org.eclipse/paho/org.eclipse.paho.client.mqttv3/1.2.5

org.eclipse.paho.client.mqttv3-1.2.5-e2artifa...	2020-07-14 13:07	1833
org.eclipse.paho.client.mqttv3-1.2.5-e2artifa...	2020-07-14 13:07	819
org.eclipse.paho.client.mqttv3-1.2.5-e2artifa...	2020-07-14 13:07	32
org.eclipse.paho.client.mqttv3-1.2.5-e2artifa...	2020-07-14 13:07	40
org.eclipse.paho.client.mqttv3-1.2.5-e2metada...	2020-07-14 13:07	4745
org.eclipse.paho.client.mqttv3-1.2.5-e2metada...	2020-07-14 13:07	819
org.eclipse.paho.client.mqttv3-1.2.5-e2metada...	2020-07-14 13:07	32
org.eclipse.paho.client.mqttv3-1.2.5-e2metada...	2020-07-14 13:07	40
org.eclipse.paho.client.mqttv3-1.2.5-pack200...	2020-07-14 13:07	91479
org.eclipse.paho.client.mqttv3-1.2.5-pack200...	2020-07-14 13:07	819
org.eclipse.paho.client.mqttv3-1.2.5-pack200...	2020-07-14 13:07	32
org.eclipse.paho.client.mqttv3-1.2.5-pack200...	2020-07-14 13:07	40
org.eclipse.paho.client.mqttv3-1.2.5-sources...	2020-07-14 13:07	245037
org.eclipse.paho.client.mqttv3-1.2.5-sources...	2020-07-14 13:07	819
org.eclipse.paho.client.mqttv3-1.2.5-sources...	2020-07-14 13:07	32
org.eclipse.paho.client.mqttv3-1.2.5-sources...	2020-07-14 13:07	40
org.eclipse.paho.client.mqttv3-1.2.5.jar	2020-07-14 13:07	241861
org.eclipse.paho.client.mqttv3-1.2.5.jar.asc	2020-07-14 13:07	819
org.eclipse.paho.client.mqttv3-1.2.5.jar.md5	2020-07-14 13:07	32
org.eclipse.paho.client.mqttv3-1.2.5.jar.sha1	2020-07-14 13:07	40
org.eclipse.paho.client.mqttv3-1.2.5.pom	2020-07-14 13:07	5977
org.eclipse.paho.client.mqttv3-1.2.5.pom.asc	2020-07-14 13:07	819
org.eclipse.paho.client.mqttv3-1.2.5.pom.md5	2020-07-14 13:07	32
org.eclipse.paho.client.mqttv3-1.2.5.pom.sha1	2020-07-14 13:07	40

* JSON-Java API 다운로드

아래 사이트에서 JSON-Java API를 다운로드한다.

<https://mvnrepository.com/>

Maven Repository: JSON-Java

Found 40593 results

Sort: **relevance** | popular | newest

1. **JSON In Java** 3,811 usages

org.json = json

JSON is a light-weight, language independent, data interchange format. See <http://www.JSON.org/> The files in this package implement JSON encoders/decoders in Java. It also includes the capability to convert between JSON and XML, HTTP headers, Cookies, and CDL. This is a reference implementation. There is a large number of JSON packages in Java. Perhaps someday the Java community will standardize on one. Until then, choose carefully. The license includes this restriction: "The software ..."

Repository	Version	Repository	Usages	Date
Central (17)	20200518	Central	106	May, 2020



JSON In Java » 20200518

JSON is a light-weight, language independent, data interchange format. See <http://www.JSON.org/> The files in this package implement JSON encoders/decoders in Java. It also includes the capability to convert between JSON and XML, HTTP headers, Cookies, and CDL. This is a reference implementation. There is a large number of JSON packages in Java. Perhaps someday the Java community will standardize on one. Until then, choose carefully. The license includes this restriction: "The software ..."

License

JSON

Categories

JSON Libraries

HomePage

<https://github.com/douglasrockford/JSON-java>

Date

(May 22, 2020)

Files

bundle (64 KB) View All

Repositories

Central

Used By

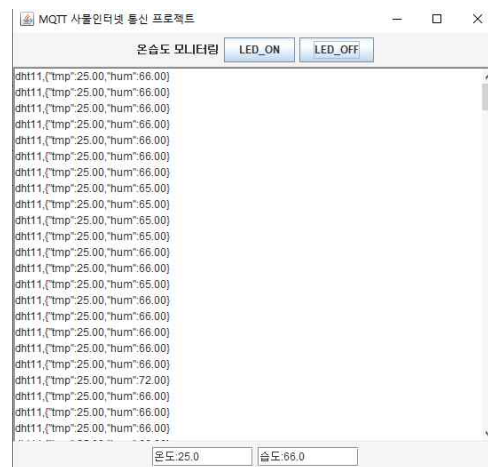
3,811 artifacts

org/json/json/20200518

...			
json-20200518-javadoc.jar	2020-05-20 00:30	165769	
json-20200518-javadoc.jar.asc	2020-05-20 00:30	473	
json-20200518-javadoc.jar.asc.md5	2020-05-20 00:30	32	
json-20200518-javadoc.jar.asc.shal	2020-05-20 00:30	40	
json-20200518-javadoc.jar.md5	2020-05-20 00:30	32	
json-20200518-javadoc.jar.shal	2020-05-20 00:30	40	
json-20200518-sources.jar	2020-05-20 00:30	77672	
json-20200518-sources.jar.asc	2020-05-20 00:30	473	
json-20200518-sources.jar.asc.md5	2020-05-20 00:30	32	
json-20200518-sources.jar.asc.shal	2020-05-20 00:30	40	
json-20200518-sources.jar.md5	2020-05-20 00:30	32	
json-20200518-sources.jar.shal	2020-05-20 00:30	40	
json-20200518.jar	2020-05-20 00:30	65966	
json-20200518.jar.asc	2020-05-20 00:30	473	
json-20200518.jar.asc.md5	2020-05-20 00:30	32	
json-20200518.jar.asc.shal	2020-05-20 00:30	40	
json-20200518.jar.md5	2020-05-20 00:30	32	
json-20200518.jar.shal	2020-05-20 00:30	40	
json-20200518.pom	2020-05-20 00:30	5974	
json-20200518.pom.asc	2020-05-20 00:30	473	
json-20200518.pom.asc.md5	2020-05-20 00:30	32	
json-20200518.pom.asc.shal	2020-05-20 00:30	40	
json-20200518.pom.md5	2020-05-20 00:30	32	
json-20200518.pom.shal	2020-05-20 00:30	40	

9.3 GUI 만들기

IoTFrame.java



```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import org.eclipse.paho.client.mqttv3.MqttMessage;
import org.json.JSONObject;

public class IoTFrame extends JFrame implements ActionListener,ReceiveEventListener{
    private static final long serialVersionUID = 1L;
    private JTextField tmp = new JTextField(10);
    private JTextField hum = new JTextField(10);
    private JButton ledOn = new JButton("LED_ON");
    private JButton ledOff = new JButton("LED_OFF");
    private JLabel msg = new JLabel("온습도 모니터링");
    private JTextArea out = new JTextArea(20,40);
    private JPanel panel = new JPanel();
    private JPanel panel1 = new JPanel();
    private JPanel panel2 = new JPanel();
    private ScrollPane sp=new ScrollPane();
    private MqttClass mqtt = null;

    public IoTFrame(MqttClass mqtt){
        this();
        this.mqtt = mqtt;
        this.mqtt.setMyEventListener(this);
    }

    public IoTFrame(){
        super("MQTT 사물인터넷 통신 프로젝트");
        setSize(400,400);
        panel.add(msg);
        panel.add(ledOn);
        panel.add(ledOff);
        panel1.add(tmp);

```

```

        panel1.add(hum);
        sp.add(out);
        add(BorderLayout.NORTH, panel); //jframe에 div
        add(BorderLayout.CENTER, sp); //jframe에 div
        add(BorderLayout.SOUTH, panel1);
        //add(BorderLayout.EAST, panel2);
        ledOn.addActionListener(this); //통지할 수 있게 설정
        ledOff.addActionListener(this); //통지할 수 있게 설정
        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    @Override
    public void recvMsg(String topic, MqttMessage msg) {
        System.out.println(topic+","+msg);
        String append = out.getText();
        out.setText(topic+","+msg+"\n"+append);
        JSONObject obj=new JSONObject(new String(msg.getPayload()));
        tmp.setText("온도:"+obj.get("tmp").toString());
        hum.setText("습도:"+obj.get("hum").toString());
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        JButton b = (JButton) e.getSource();
        if(b.getText().equals("LED_ON")){
            mqtt.sendMessage("1");
        }else if(b.getText().equals("LED_OFF")) {
            mqtt.sendMessage("2");
        }
    }
}

```

9.4 mosquitto 연결 데이터 통신

MqttClass.java

```

import java.util.UUID;
import org.eclipse.paho.client.mqttv3.*;
public class MqttClass implements MqttCallback{

    private MqttClient client = null;
    public MqttClass(){
        new Thread(task1).start();
    }
    private ReceiveEventListener listener = null;

    Runnable task1 = new Runnable(){
        @Override
        public void run() {
            try {
                String clientId = UUID.randomUUID().toString();
                //new MqttClient()
                client = new MqttClient("tcp://172.30.1.44:1883", clientId);
                MqttConnectOptions connopt = new MqttConnectOptions();
                connopt.setCleanSession(true);
                client.connect(connopt);
                client.setCallback(MqttClass.this);
                client.subscribe("dht11");

                new IoTFrame(MqttClass.this);

            } catch (MqttException e) {
                System.out.println("ERROR"+e.getStackTrace());
            }
        }
    };

    public void sendMessage(String payload){
        MqttMessage message = new MqttMessage();
        message.setPayload(payload.getBytes());
        try {
            if(client.isConnected()){
                client.publish("led", message);
            }
        } catch (MqttException e) {
            System.out.println("error1-"+e.getStackTrace());//+e.getMessage());
        }
    }
}

```

```

@Override
public void connectionLost(Throwable arg0) {
    try {
        System.out.println("disconnect");
        client.close();
    } catch (MqttException e) {
        System.out.println("error"+e.getMessage());
    }
}

@Override
public void deliveryComplete(IMqttDeliveryToken arg0) {

}

public void setMyEventListener(ReceiveEventListener listener){
    this.listener = listener;
}

@Override
public void messageArrived(String topic, MqttMessage msg) throws Exception {
    //System.out.println(topic+", "+msg.toString());
    listener.recvMsg(topic, msg);
}
}

```

- 인터페이스 만들기

ReceiveEventListener.java

```

import org.eclipse.paho.client.mqttv3.MqttMessage;

public interface ReceiveEventListener {
    public void recvMsg(String topic, MqttMessage msg);

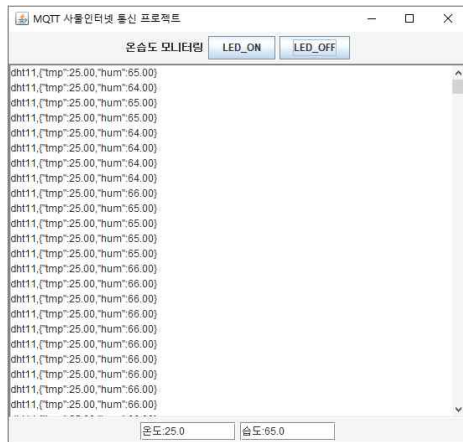
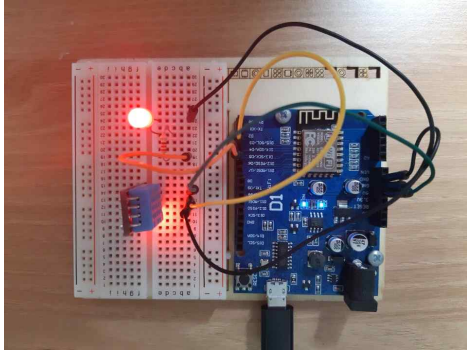
}

```

9.5 main 클래스 만들기 및 구동

JavaMQTT.java

```
public class JavaMQTT{
    public static void main(String[] args) {
        new MqttClass();
    }
}
```



C:\Program Files\mosquitto>mosquitto_sub -d -t led

```
Client mosq-09PxEFFcni0qzHdnq sending PINGREQ
Client mosq-09PxEFFcni0qzHdnq received PINGRESP
Client mosq-09PxEFFcni0qzHdnq sending PINGREQ
Client mosq-09PxEFFcni0qzHdnq received PINGRESP
Client mosq-09PxEFFcni0qzHdnq sending PINGREQ
Client mosq-09PxEFFcni0qzHdnq received PINGRESP
Client mosq-09PxEFFcni0qzHdnq received PUBLISH (d0, q0, r0, m0, 'led', ... (1 bytes))
1
Client mosq-09PxEFFcni0qzHdnq received PUBLISH (d0, q0, r0, m0, 'led', ... (1 bytes))
2
Client mosq-09PxEFFcni0qzHdnq received PUBLISH (d0, q0, r0, m0, 'led', ... (1 bytes))
1
Client mosq-09PxEFFcni0qzHdnq received PUBLISH (d0, q0, r0, m0, 'led', ... (1 bytes))
2
```

C:\Program Files\mosquitto>mosquitto_sub -d -t dht11

```
Client mosq-HL78qAA3ZVLorTaPa1 sending PINGREQ
Client mosq-HL78qAA3ZVLorTaPa1 sending CONNECT
Client mosq-HL78qAA3ZVLorTaPa1 received CONNACK (0)
Client mosq-HL78qAA3ZVLorTaPa1 sending SUBSCRIBE (Mid: 2, Topic: dht11, QoS: 0, Options: 0x00)
Client mosq-HL78qAA3ZVLorTaPa1 received SUBACK
Subscribed (mid: 2): 0
Client mosq-HL78qAA3ZVLorTaPa1 received PUBLISH (d0, q0, r0, m0, 'dht11', ... (25 bytes))
{"tmp":25.00,"hum":66.00}
Client mosq-HL78qAA3ZVLorTaPa1 received PUBLISH (d0, q0, r0, m0, 'dht11', ... (25 bytes))
{"tmp":25.00,"hum":66.00}
Client mosq-HL78qAA3ZVLorTaPa1 received PUBLISH (d0, q0, r0, m0, 'dht11', ... (25 bytes))
{"tmp":25.00,"hum":66.00}
Client mosq-HL78qAA3ZVLorTaPa1 received PUBLISH (d0, q0, r0, m0, 'dht11', ... (25 bytes))
{"tmp":25.00,"hum":65.00}
Client mosq-HL78qAA3ZVLorTaPa1 received PUBLISH (d0, q0, r0, m0, 'dht11', ... (25 bytes))
{"tmp":25.00,"hum":65.00}
Client mosq-HL78qAA3ZVLorTaPa1 received PUBLISH (d0, q0, r0, m0, 'dht11', ... (25 bytes))
{"tmp":25.00,"hum":65.00}
```