

1 다운, 환경설정

2018년 10월 29일 월요일 오전 10:05

genymotion



전체

동영상

이미지

뉴스

지도

더보기

설정

검색결과 약 841,000개 (0.20초)

Genymotion Android Emulator | Cloud-based Android virtual devices ...

<https://www.genymotion.com/> ▼ 이 페이지 번역하기

Cross-platform Android emulator for developers & QA engineers. Develop & automate your tests to

Start a new Android Studio project

Select the form factors and minimum SDK

Some devices require additional SDKs. Low API levels target more devices, but offer fewer API features.

☒ Phone and Tablet

API 18: Android 4.3 (Jelly Bean)

By targeting API 18 and later, your app will run on approximately 95.9% of devices. [Help me choose](#)

☐ Include Android Instant App support

☐ Wear OS

API 23: Android 6.0 (Marshmallow)

☐ TV

API 21: Android 5.0 (Lollipop)

☐ Android Auto

☐ Android Things

API 24: Android 7.0 (Nougat)

Previous

Next

Cancel

Finish



Add an Activity to Mobile



Add No Activity



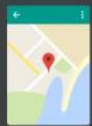
Basic Activity

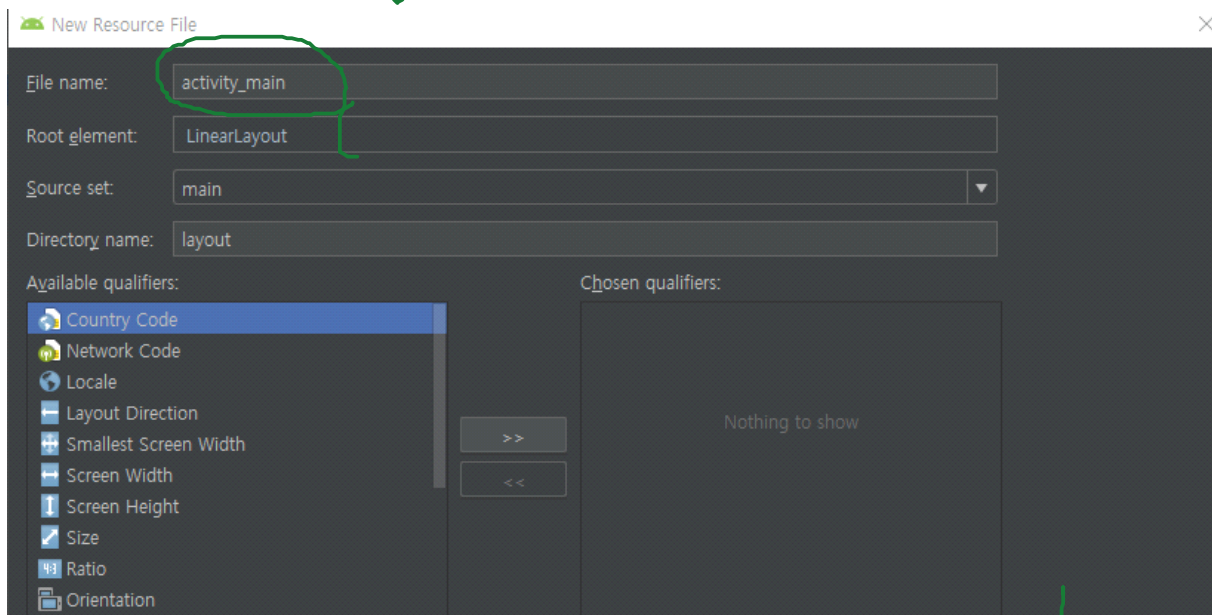
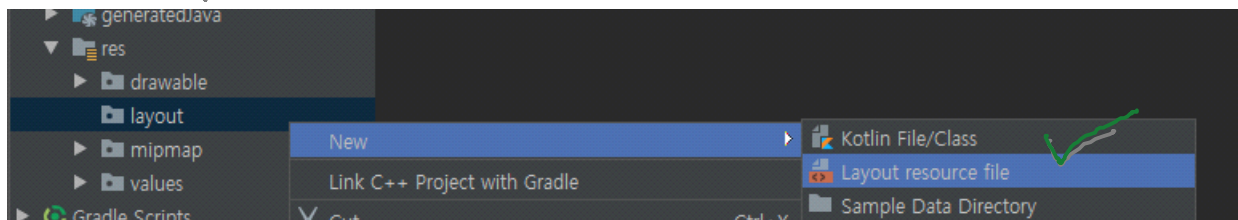
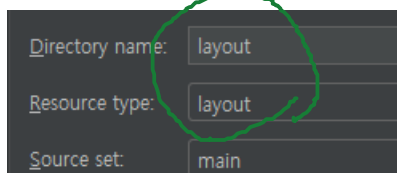
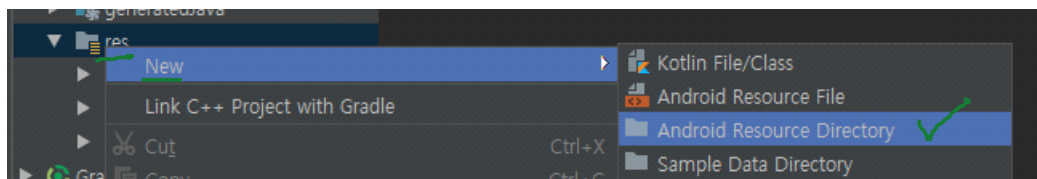
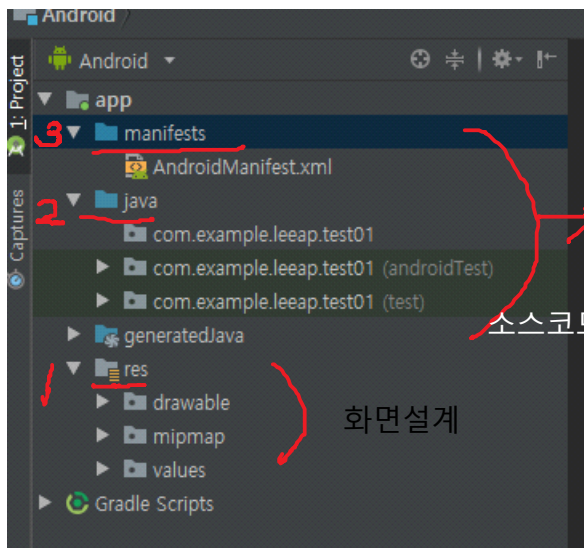


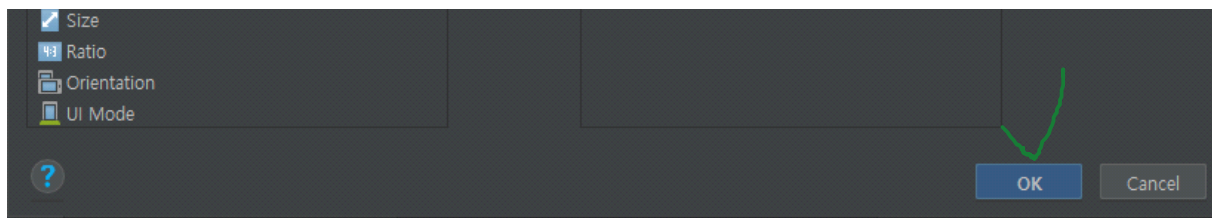
Bottom Navigation Activity



Empty Activity

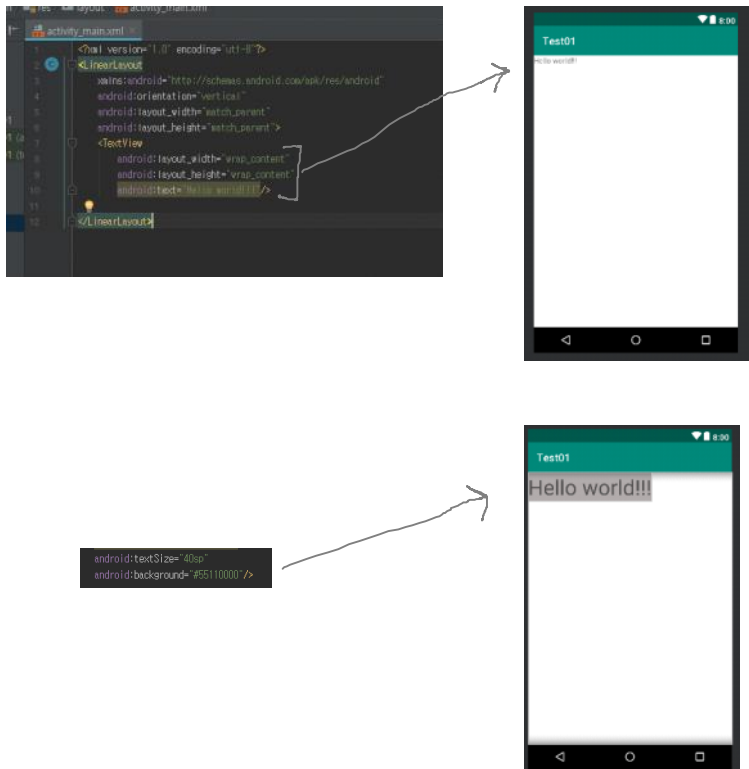


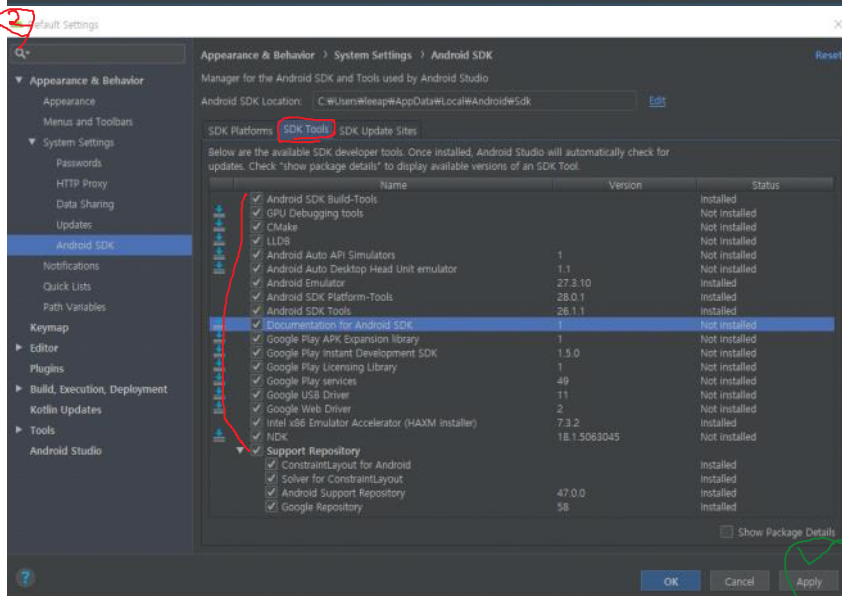
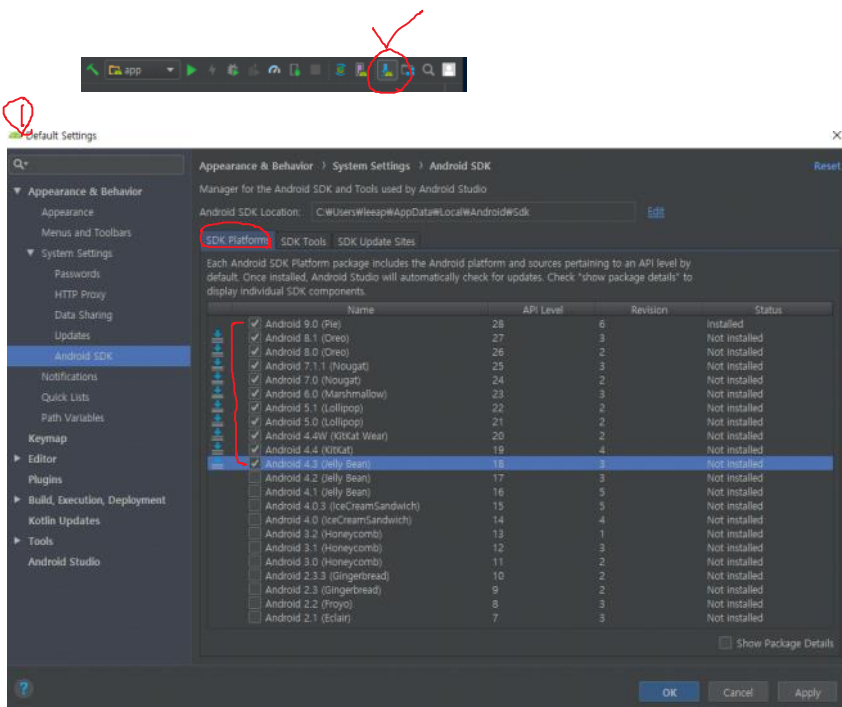




2. test 기본

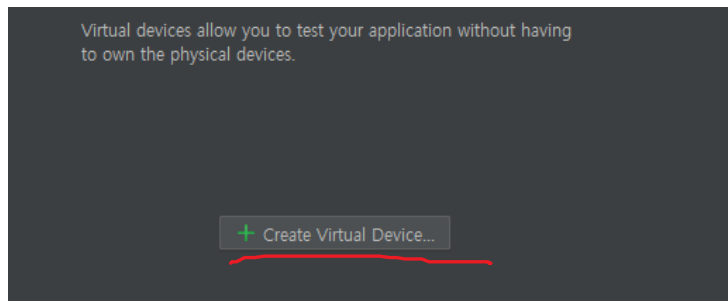
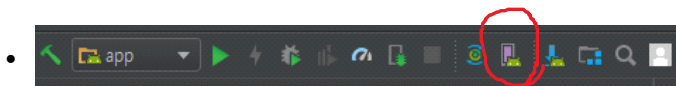
2018년 10월 29일 월요일 오전 11:10






3. 기본 앱 구성

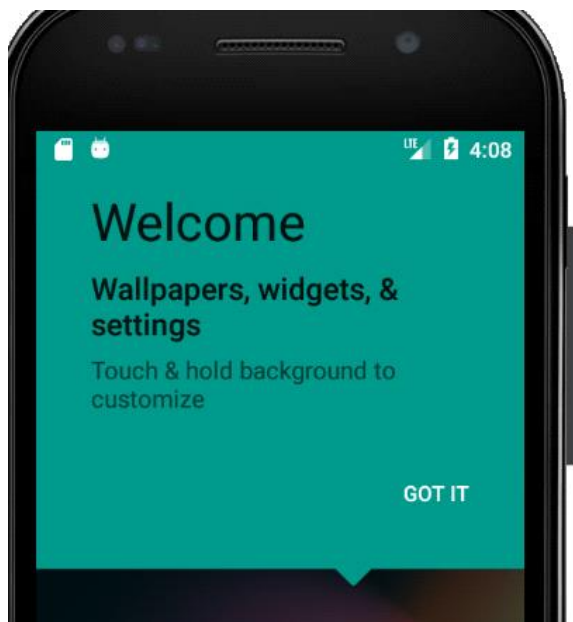
2018년 10월 29일 월요일 오후 1:00



| Category | Name | Play Store | Screen size |
|----------|------------|------------|-------------|
| TV | Pixel XL | | 5.5" |
| Phone | Pixel 2 XL | | 5.99" |
| Wear OS | Pixel 2 | | 5.0" |
| Tablet | Pixel | | 5.0" |
| | Nexus S | | 4.0" |
| | Nexus One | | 3.7" |

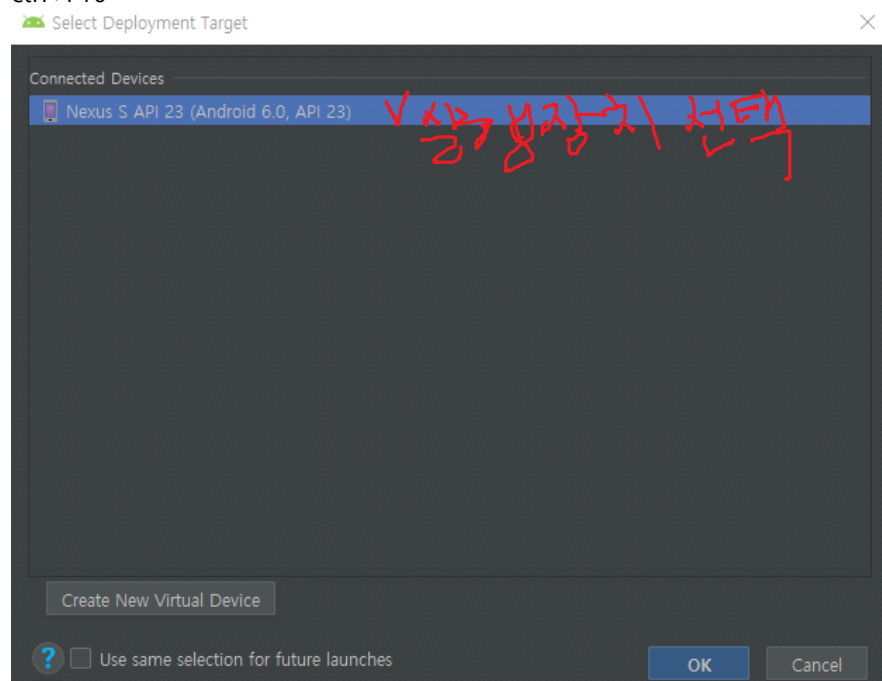


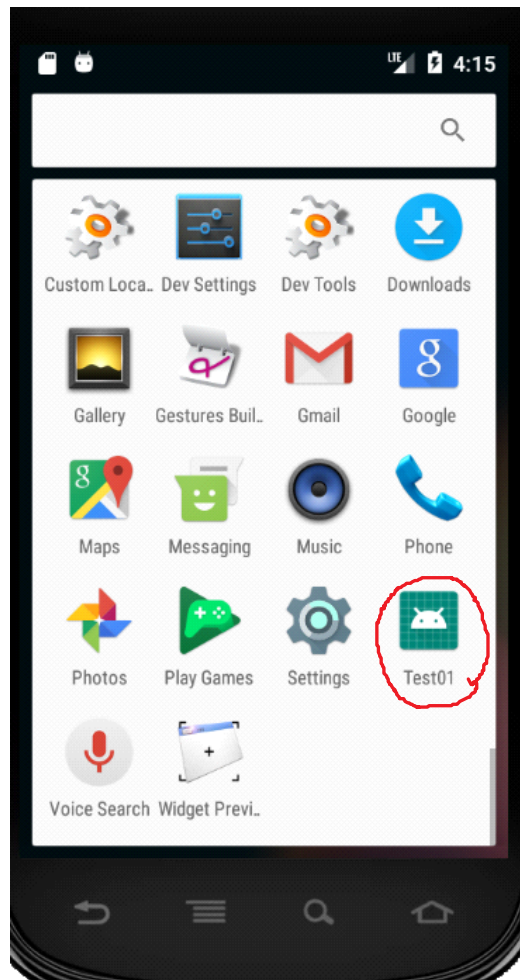
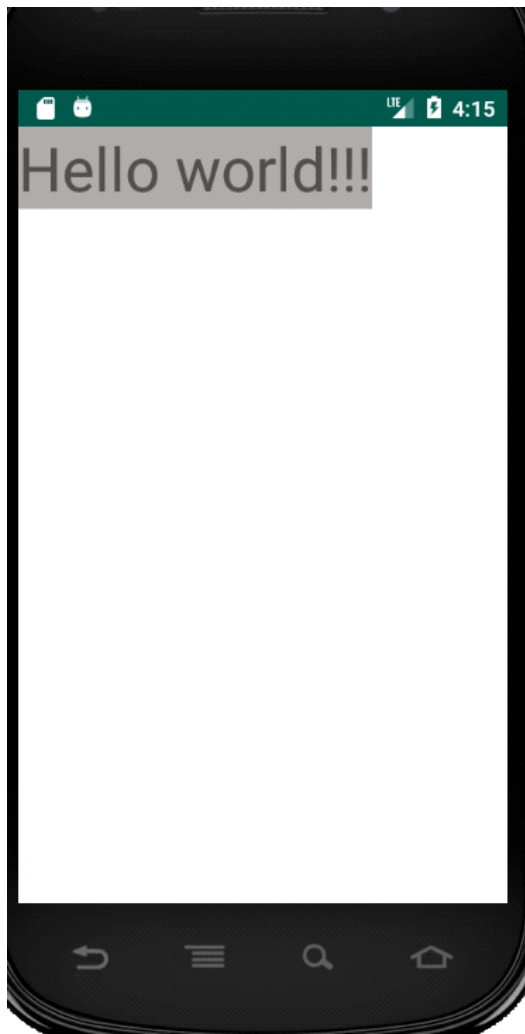
| Type | Name | Play Store | Resolution | API | Target | CPU/ABI | Size on Disk | Actions |
|------|----------------|------------|-----------------|-----|---------------------------|---------|--------------|---|
| | Nexus S API 23 | | 480 x 800: hdpi | 23 | Android 6.0 (Google AP... | x86 | 2.5 GB |  |





ctrl+F10

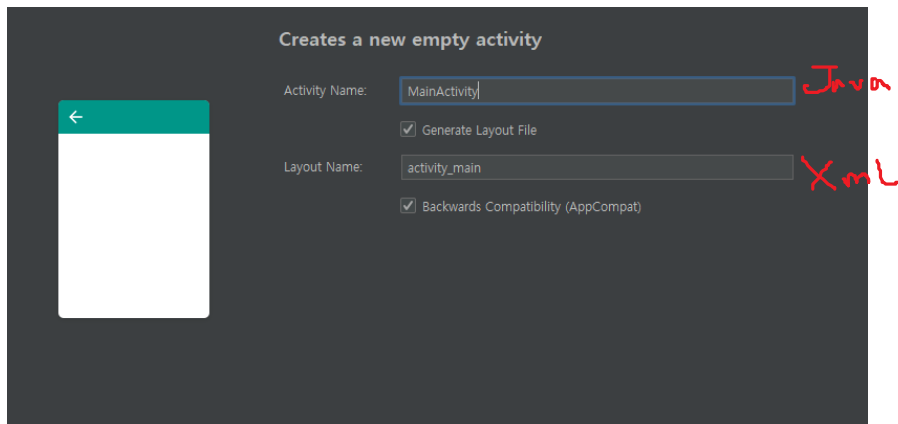




4. Test02 (Toast)

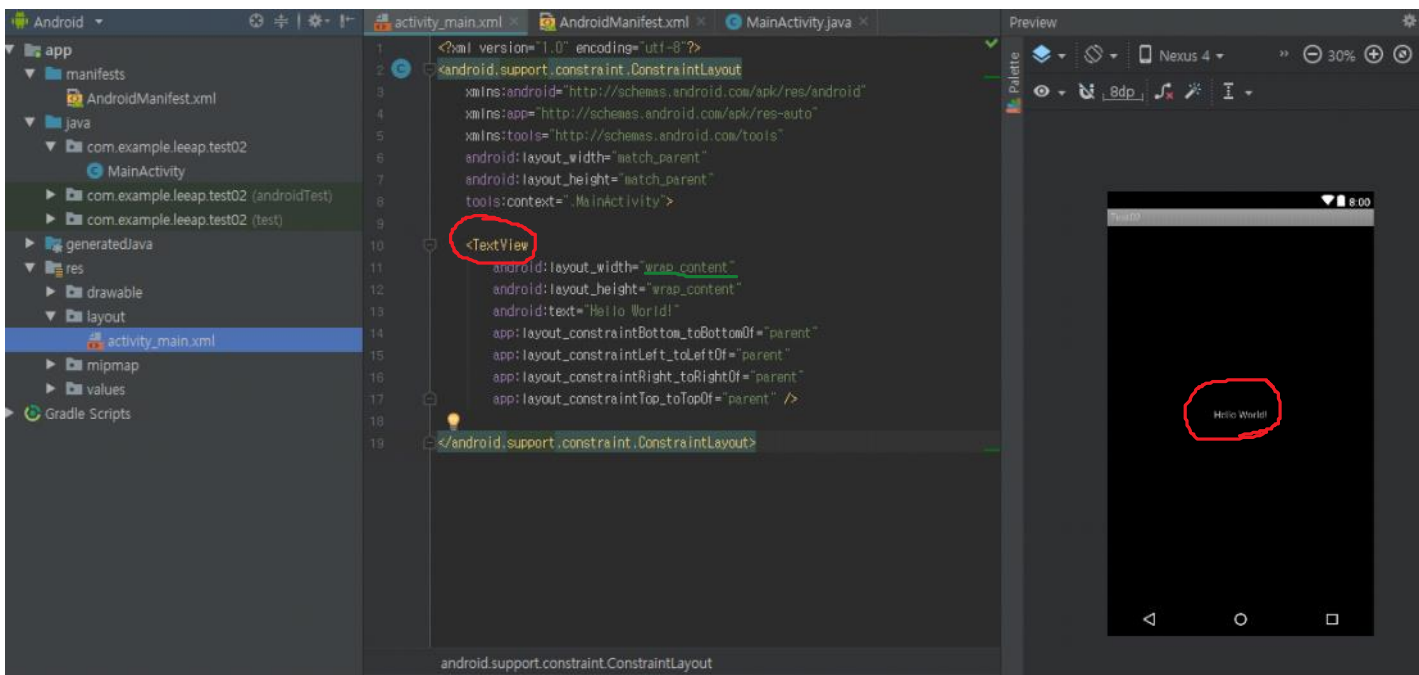
2018년 10월 30일 화요일 오전 9:11

ConstraintLayout : 4가지 방향으로 배치를 하는 레이아웃



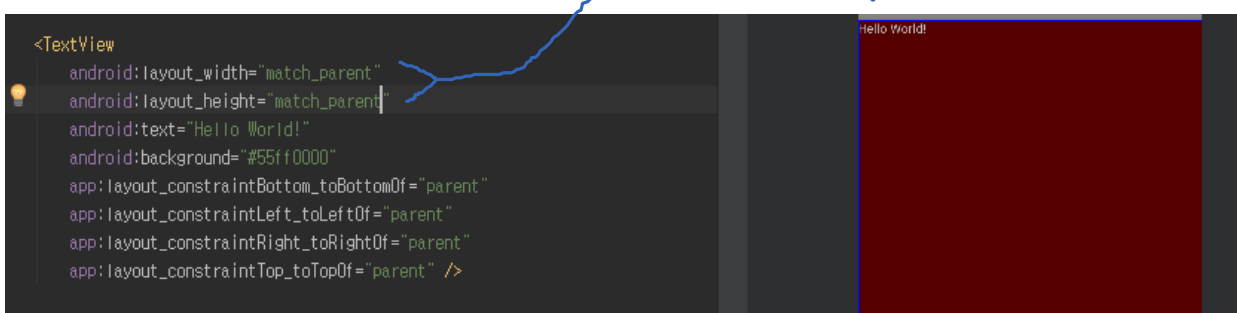
레이아웃 : 일반 뷰 그룹

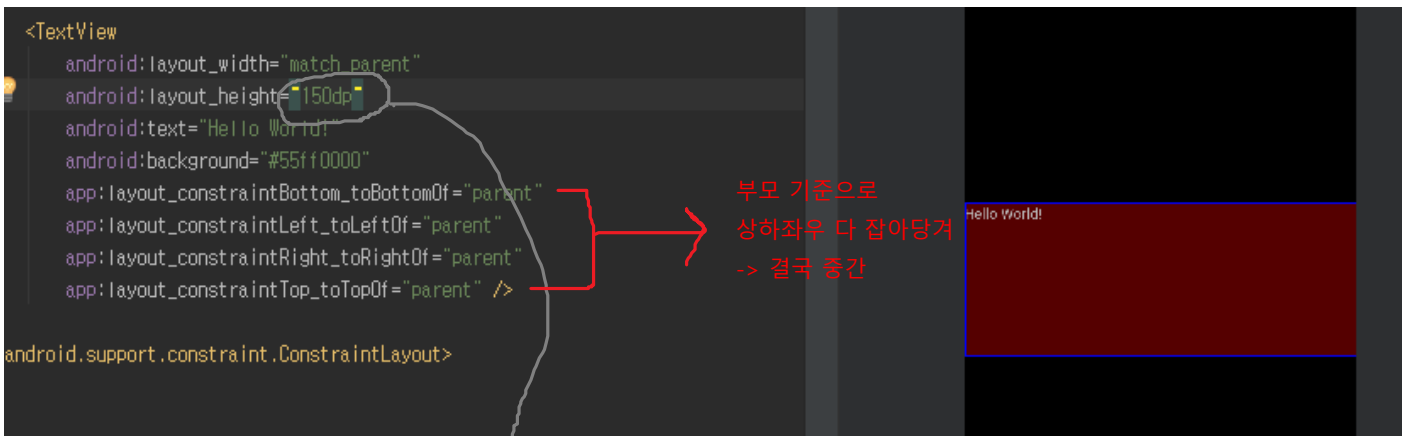
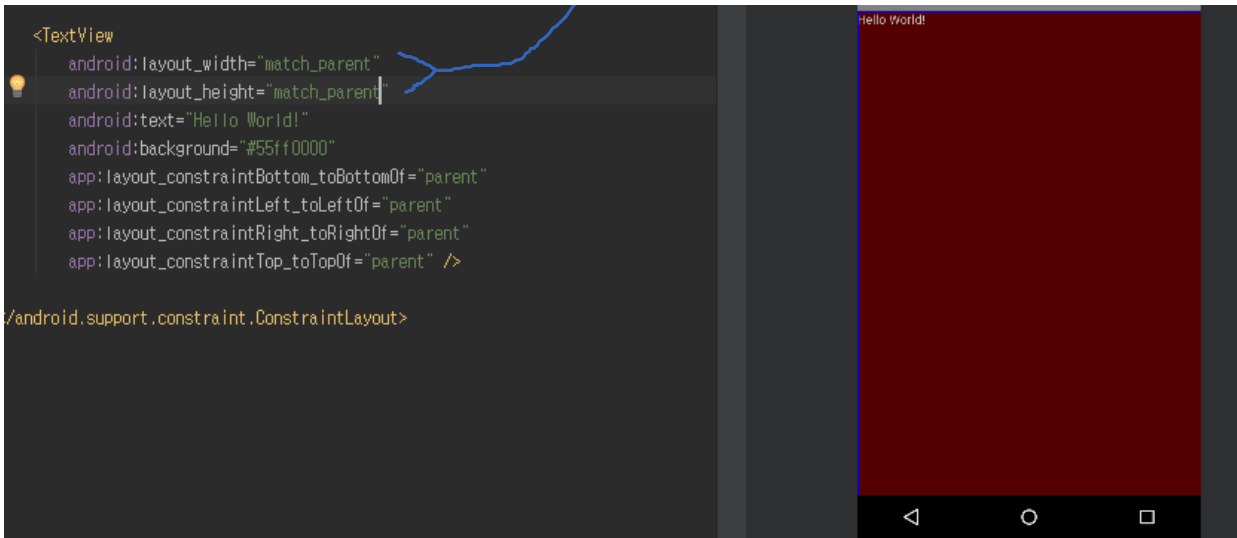
일반 뷰 : 버튼, 입력, ... 기능



✓ Wrap content 필요한 부분만쓰고 비워라

Parent에 맞춰 다 채우는 것

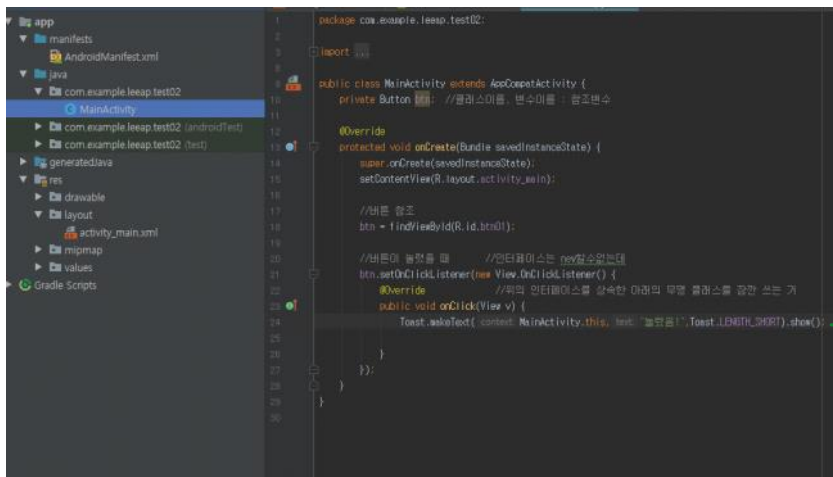




Device에 따라 크기가 다르니까, 어떤 크기던 뭐 100칸씩 나눠서 150만큼채워라~ 이런



Text_hello id를 가진 값의 top에
버튼의 bottom을 맞춰라



```

private Button btn; //클래스이름, 변수이름 : 참조변수
private TextView textHello;

```

```

//버튼 참조
btn = findViewById(R.id.btn01);
textHello = findViewById(R.id.text_hello);

```

```

//버튼이 눌렸을 때 //인터페이스는 new할수없는데
btn.setOnClickListener(new View.OnClickListener() {
    @Override //위의 인터페이스를 상속한 아래의 무명 클래스를 잠깐 쓰는 거
    public void onClick(View v) {
        Toast.makeText(context, MainActivity.this, text, "눌렀음!", Toast.LENGTH_SHORT).show();
        textHello.setText("눌렀다니까요~~");
    }
});

```



예제 1 (카운트업)

2018년 10월 30일 화요일 오전 10:40

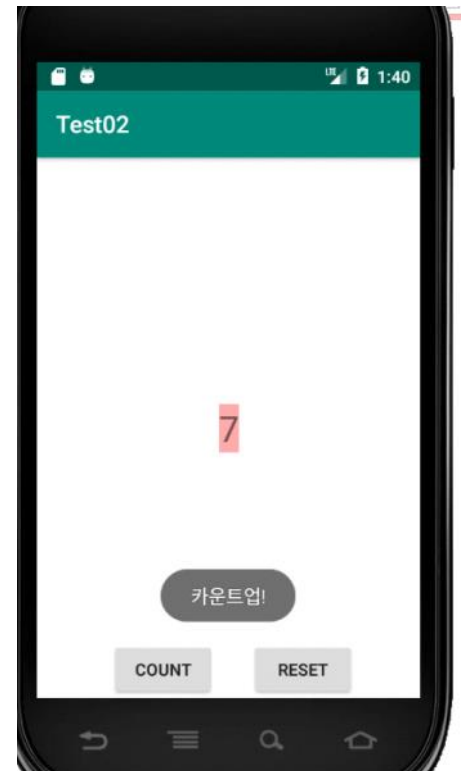
내 코드

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

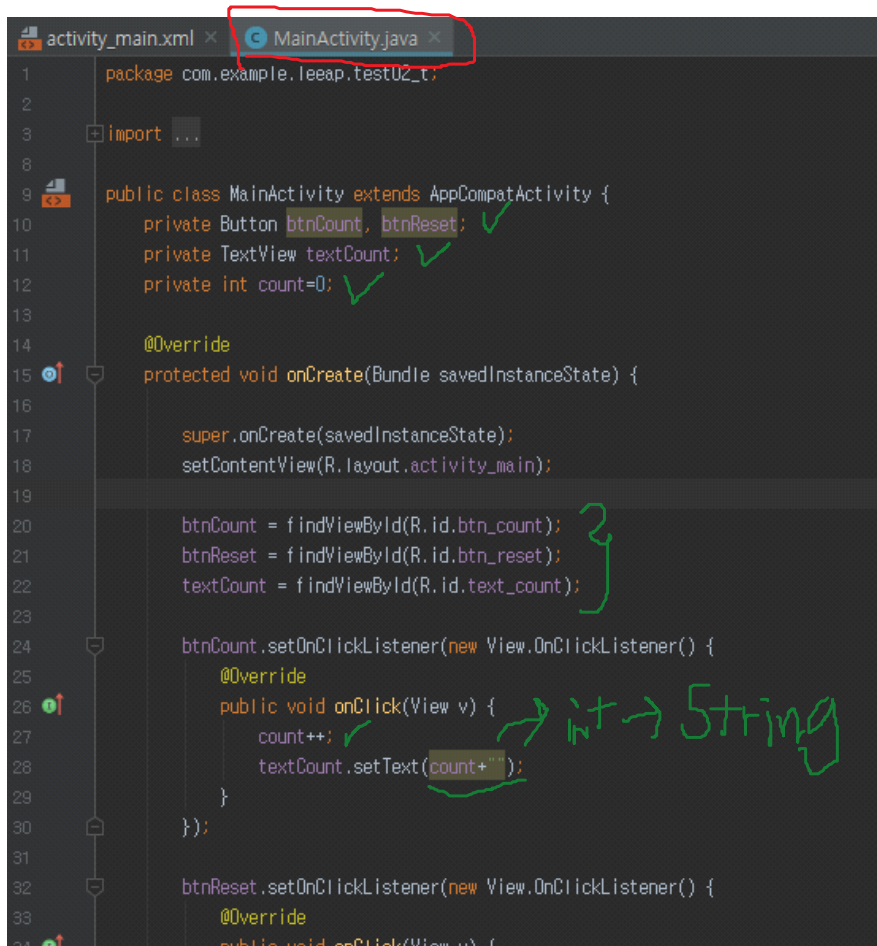
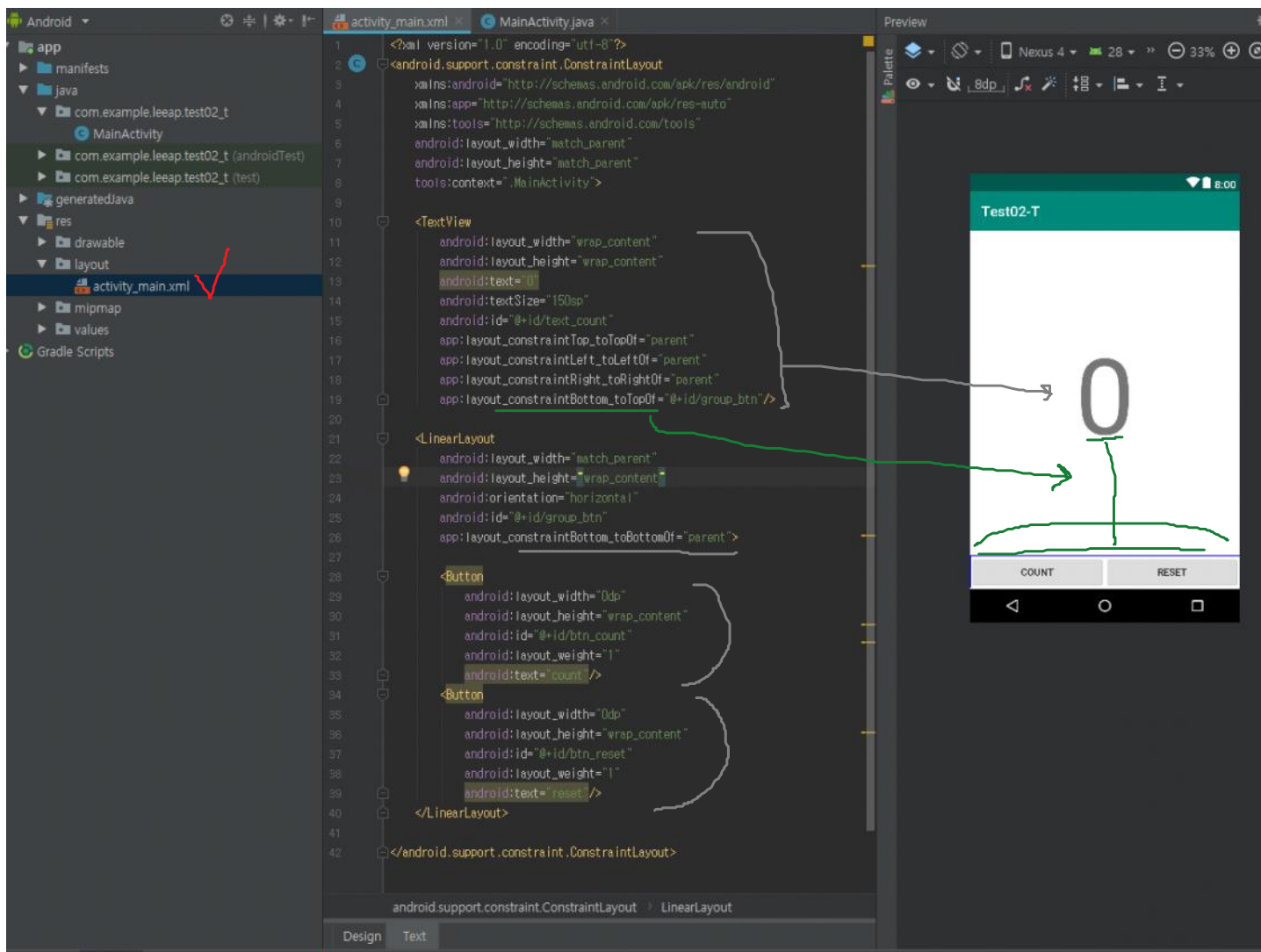
    //버튼 참조
    btn1 = findViewById(R.id.btn01);
    btn2 = findViewById(R.id.btn02);
    textHello = findViewById(R.id.text_hello);

    //버튼이 눌렀을 때 //인터페이스는 new할수없는데
    btn1.setOnClickListener(new View.OnClickListener() {
        @Override //위의 인터페이스를 상속한 아래의 무명 클래스를 잠깐 쓰는 거
        public void onClick(View v) {
            int cnt = Integer.parseInt(textHello.getText().toString());
            String result = String.valueOf(cnt+1);
            Toast.makeText(context: MainActivity.this, text: "카운트업!", Toast.LENGTH_SHORT).show();
            textHello.setText(result);
        }
    });

    //버튼이 눌렀을 때 //인터페이스는 new할수없는데
    btn2.setOnClickListener(new View.OnClickListener() {
        @Override //위의 인터페이스를 상속한 아래의 무명 클래스를 잠깐 쓰는 거
        public void onClick(View v) {
            Toast.makeText(context: MainActivity.this, text: "초기화!", Toast.LENGTH_SHORT).show();
            textHello.setText("0");
        }
    });
}
```



선생님 코드

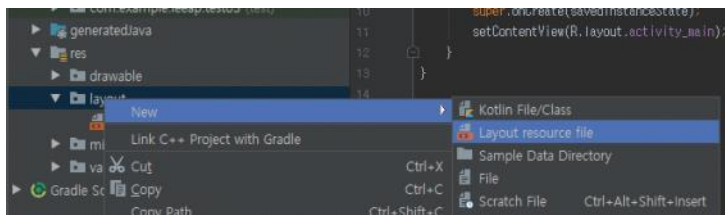


```
31
32 btnReset.setOnClickListener(new View.OnClickListener() {
33     @Override
34     public void onClick(View v) {
35         count=0;
36         textCount.setText(count+"");
37     }
38 });
39
40 }
41
42
```

5. Test03 -이벤트처리

2018년 10월 31일 수요일 오전 9:21

Test03



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <CheckBox
        android:id="@+id/check_meat"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="고기" />

    <CheckBox
        android:id="@+id/check_cheese"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="치즈" />

    <CheckBox
        android:id="@+id/check_bacon"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="베이컨" />

    <RadioGroup
        android:id="@+id/radioGroup"
        android:orientation="horizontal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">

        <RadioButton
            android:id="@+id/radio_red"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="빨강" />

        <RadioButton
            android:id="@+id/radio_blue"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="파랑" />

    </RadioGroup>

    <RatingBar
        android:id="@+id/rating"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:isIndicator="false"
        android:numStars="5"
        android:rating="0"
        android:stepSize="0.5">

    <ToggleButton
        android:id="@+id/btn_toggle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textOn="켜짐"
        android:textOff="꺼짐" />

    <EditText
        android:id="@+id/edit_enter"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="입력후 엔터" />

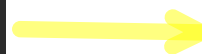
</LinearLayout>
```

```
<RatingBar
    android:id="@+id/rating"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:isIndicator="false"
    android:numStars="5"
    android:rating="0"
    android:stepSize="0.5">

<ToggleButton
    android:id="@+id/btn_toggle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textOn="켜짐"
    android:textOff="꺼짐" />

<EditText
    android:id="@+id/edit_enter"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="입력후 엔터" />

</LinearLayout>
```



d

checkBox

```
public class MainActivity extends AppCompatActivity {

    private CheckBox checkMeat, checkCheese, checkBacon;
    private RadioButton radioRed, radioBlue;
    private RatingBar ratingBar;
    private EditText editEnter;
    private ToggleButton toggleButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main); //inflate

        checkMeat = findViewById(R.id.check_meat);
        checkCheese = findViewById(R.id.check_cheese);
        checkBacon = findViewById(R.id.check_bacon);
        radioRed = findViewById(R.id.radio_red);
        radioBlue = findViewById(R.id.radio_blue);
        ratingBar = findViewById(R.id.rating);
        editEnter = findViewById(R.id.edit_enter);
        toggleButton = findViewById(R.id.btn_toggle);

        //alt+enter : import단축키
        public void onCheck(View v){
        }
    }
}
```

이거 바꿔줘야 두번째 layout이 실행됨

```
<CheckBox
    android:id="@+id/check_meat"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="고기"
    android:onClick="onCheck" />

<CheckBox
    android:id="@+id/check_cheese"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="치즈"
    android:onClick="onCheck" />
```

여러 개의 자식객체의 주소를 저장하는 View

```
//alt+enter : import(단축키)
public void onCheck(View v){
}
}
```

여러 개의 자식객체의 주소를 저장하는 View

```
<CheckBox
    android:id="@+id/check_meat"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="고기"
    android:onClick="onCheck"/>

<CheckBox
    android:id="@+id/check_cheese"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="치즈"
    android:onClick="onCheck"/>

<CheckBox
    android:id="@+id/check_bacon"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="베이컨"
    android:onClick="onCheck"/>
```

```
//alt+enter : import(단축키)
public void onCheck(View v){
    CheckBox checked = (CheckBox)v; //자식클래스로 형변환
    String str = "";
    //지금 체크된 놈 아이디
    switch (checked.getId()){
        case R.id.check_meat:
            str += "고기";
            break;
        case R.id.check_bacon:
            str += "베이컨";
            break;
        case R.id.check_cheese:
            str += "치이즈";
            break;
    }

    if(checked.isChecked()){
        Toast.makeText( context this, text str+" 체크됨", Toast.LENGTH_SHORT).show();
    }else{
        Toast.makeText( context this, text str+" 해제됨", Toast.LENGTH_SHORT).show();
    }
}
```



Radio

```
//클래스 만들기 ! 이를 붙인 클래스~ !
class RedBlueEvent implements View.OnClickListener {
}

MainActivity -> RedBlueEvent
```

```
//클래스 만들기 ! 이를 붙인 클래스~ !
class RedBlueEvent implements View.OnClickListener {

    @Override
    public void onClick(View v) {
        RadioButton checked = (RadioButton)v;
        String select = "";
        if(checked.getId() == R.id.radio_red)
            select += "빨강 선택됨";
        else if(checked.getId() == R.id.radio_blue)
            select += "파랑 선택됨";

        Toast.makeText( context: MainActivity.this, select, Toast.LENGTH_SHORT).show();
    }
}
```

상위에 코드 추가

```
public class MainActivity extends AppCompatActivity {

    private CheckBox checkMeat, checkCheese, checkBacon;
    private RadioButton radioRed, radioBlue;
    private RatingBar ratingBar;
    private EditText editEnter;
    private ToggleButton toggleButton;

    private RedBlueEvent radioEvent = new RedBlueEvent(); //이벤트 처리기 객체
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main2); //inflate

    checkMeat = findViewById(R.id.check_meat);
    checkCheese = findViewById(R.id.check_cheese);
    checkBacon = findViewById(R.id.check_bacon);
    radioRed = findViewById(R.id.radio_red);
    radioBlue = findViewById(R.id.radio_blue);
    ratingBar = findViewById(R.id.rating);
}
```



```
private RatingBar ratingBar;
private EditText editEnter;
private ToggleButton toggleButton;

private RedBlueEvent radioEvent = new RedBlueEvent(); //이벤트 처리기 객체
```

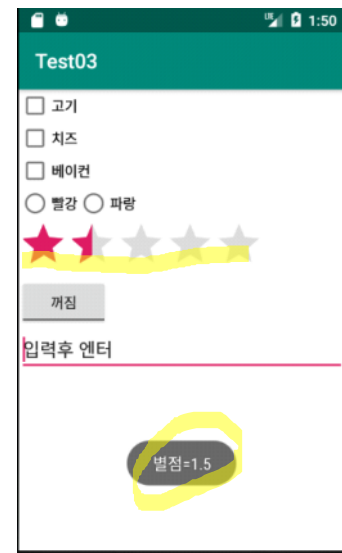
```
checkCheese = findViewById(R.id.check_cheese);
CheckBacon = findViewById(R.id.check_bacon);
radioRed = findViewById(R.id.radio_red);
radioBlue = findViewById(R.id.radio_blue);
ratingBar = findViewById(R.id.rating);
editEnter = findViewById(R.id.edit_enter);
toggleButton = findViewById(R.id.btn_toggle);

radioRed.setOnClickListener(radioEvent);
radioBlue.setOnClickListener(radioEvent);
}
```

Rating

```
//점수를 바꿨을때 -> ratingBar에 변경이 일어났을때. //-> 무명클래스
ratingBar.setOnRatingBarChangeListener(new RatingBar.OnRatingBarChangeListener() {
    @Override
    public void onRatingChanged(RatingBar ratingBar, float rating, boolean fromUser) {
    }
});
```

```
//점수를 바꿨을때 -> ratingBar에 변경이 일어났을때. //-> 무명클래스
ratingBar.setOnRatingBarChangeListener(new RatingBar.OnRatingBarChangeListener() {
    @Override
    //사용자가 눌러서 바꿨을때 fromUser = true
    public void onRatingChanged(RatingBar ratingBar, float rating, boolean fromUser) {
        Toast.makeText(context, MainActivity.this, text, "별점="+rating, Toast.LENGTH_SHORT).show();
    }
});
```



Toast

```
Toast.makeText(context, MainActivity.this, text, "별점="+rating, Toast.LENGTH_SHORT).show();
```

```
Toast.makeText(context, this, text, str+ " 체크됨", Toast.LENGTH_SHORT).show();
```

액티비티에 주소를 줄 때, main액티비티의 클래스일때는 그냥 this해도 알아듣는데 무명클래스의 경우 MainActivity.this!

내가 소속된 액티비티를 명확히 모를 때 -> `getApplicationContext()`

Text

```
editEnter.setOnKeyListener(new View.OnKeyListener() {
    @Override
    public boolean onKey(View v, int keyCode, KeyEvent event) {
        if (keyCode == KeyEvent.KEYCODE_ENTER) {
            String input = editEnter.getText().toString();
            Toast.makeText(context, MainActivity.this, text, "입력된내용: "+input, Toast.LENGTH_SHORT).show();
        }
        return false;
    }
});
```

```
<EditText
    android:id="@+id/edit_enter"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="입력후 엔터"
    android:inputType="text"
    android:maxLength="1"
/>
```



```

<EditText
    android:id="@+id/edit_enter"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="입력후 엔터"
    android:inputType="text"
    android:maxLines="1"
/>

```

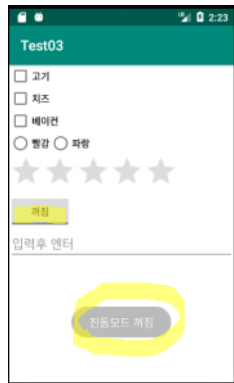


Toggle Button

```

toggleButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        ToggleButton clicked = (ToggleButton) v;
        if(clicked.isChecked()){
            Toast.makeText( context: MainActivity.this, text: "진동모드 켜짐", Toast.LENGTH_SHORT).show();
        }else{
            Toast.makeText( context: MainActivity.this, text: "진동모드 꺼짐", Toast.LENGTH_SHORT).show();
        }
    }
});

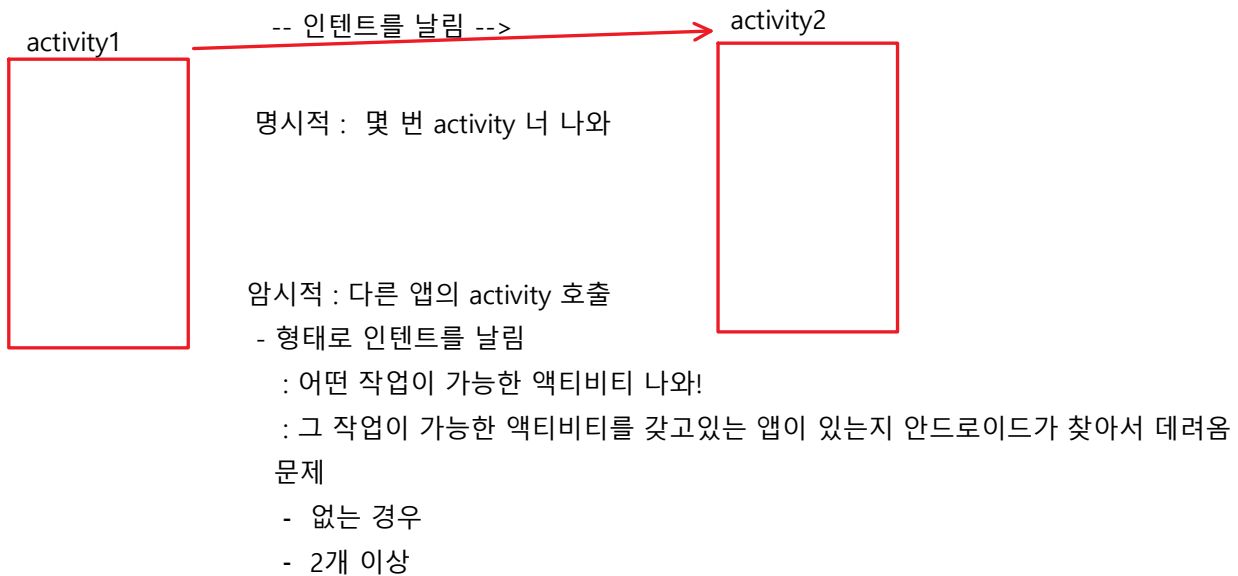
```



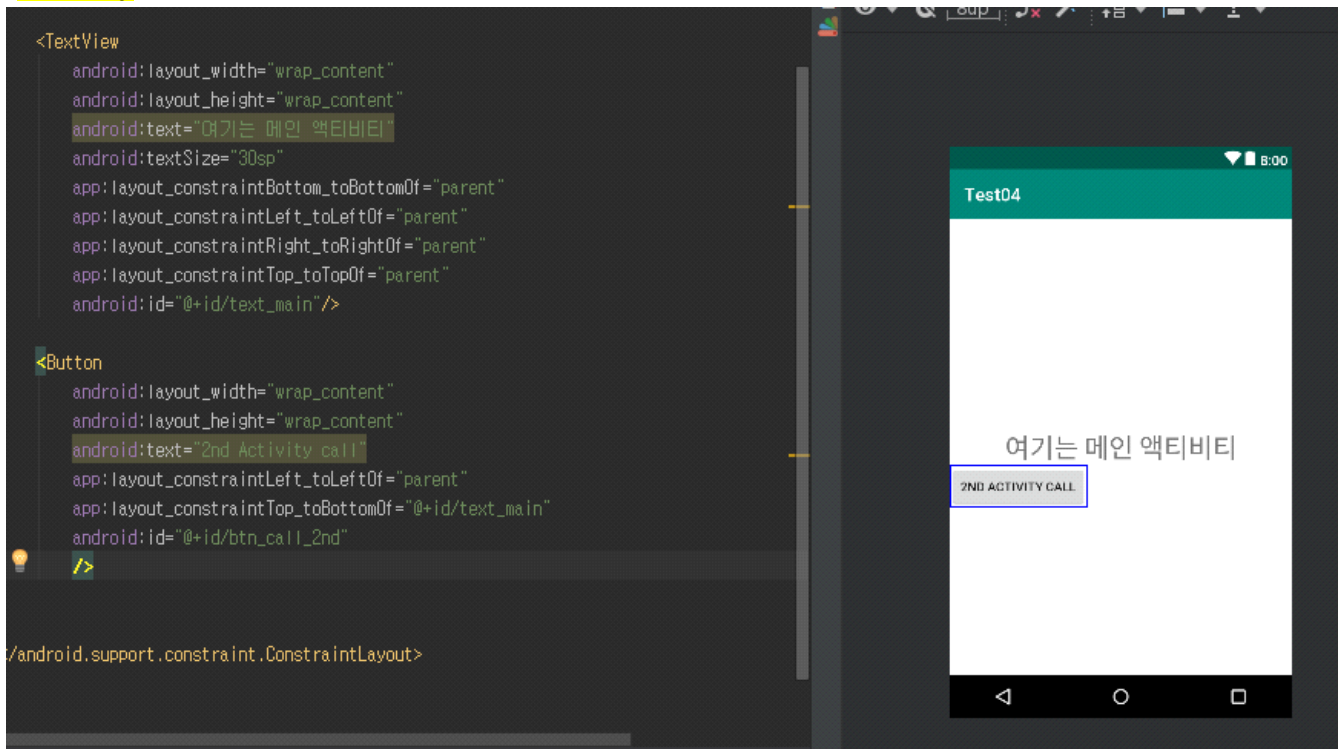
5. Test04- activity 호출

2018년 11월 1일 목요일 오전 9:26

1. Layout에 버튼 만들기
2. Layout파일만들기
3. 자바파일 만들기(연결액티비티, 인텐트보내기액티비티)
4. Manifest.xml에 액티비티 추가



1 Activity

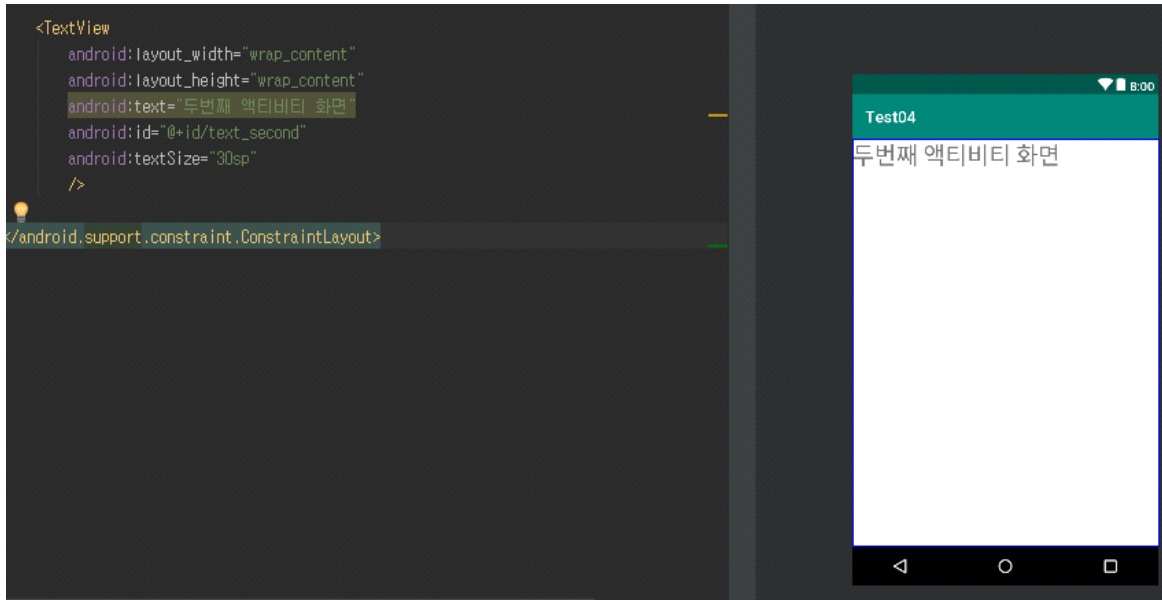


2nd Activity 생성

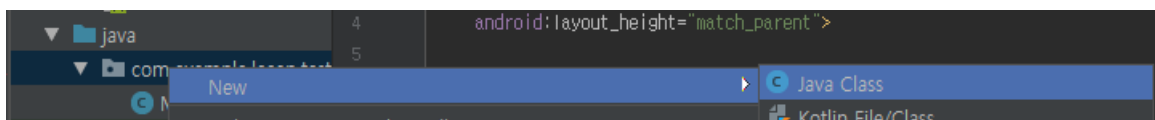
New Resource File

File name:

Root element:



SecondActivity java class 만들기



Create New Class

Name:

Kind: ☐ Class

Superclass:

Interface(s):

Package:

Visibility: ☒ Public ☐ Package Private

Modifiers: ☒ None ☐ Abstract ☐ Final

☐ Show Select Overrides Dialog

OK Cancel

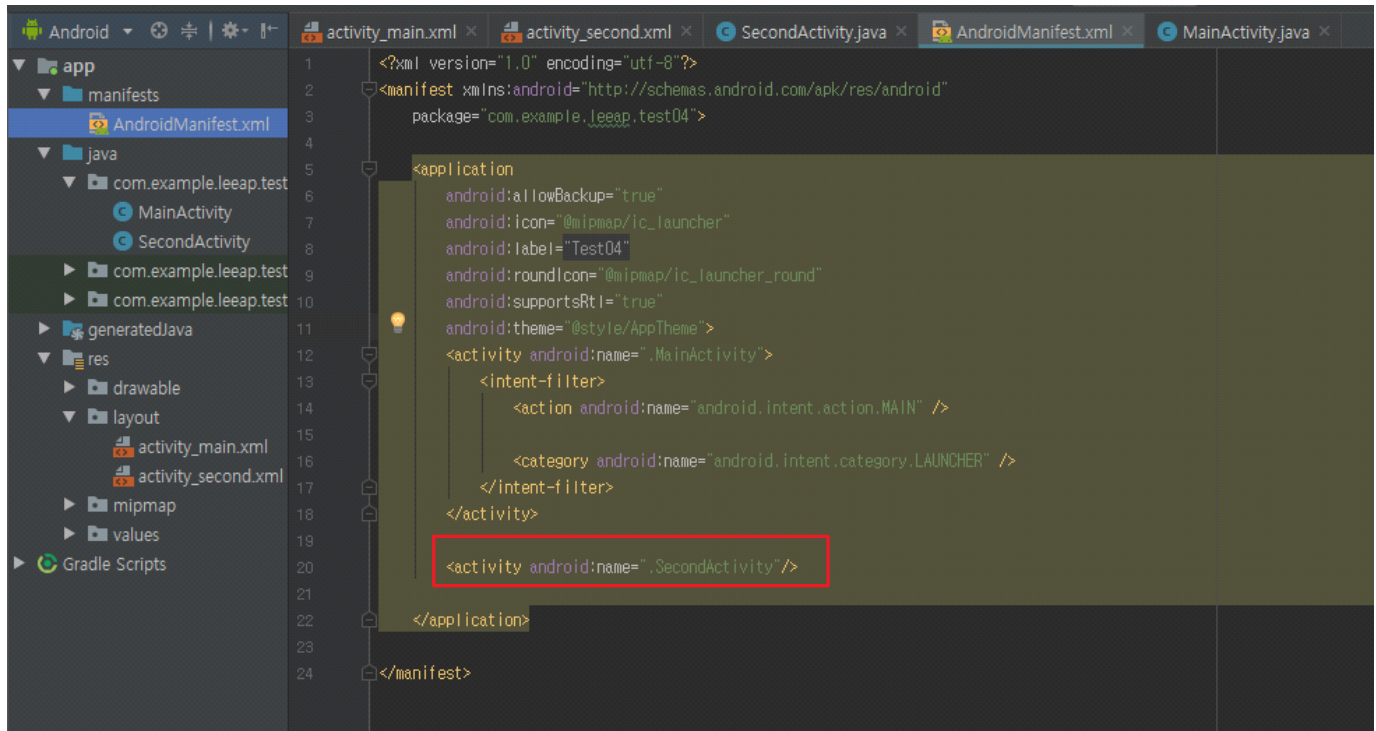
```
package com.example.leeap.test04;

import android.app.Activity;
import android.os.Bundle;

public class SecondActivity extends Activity {

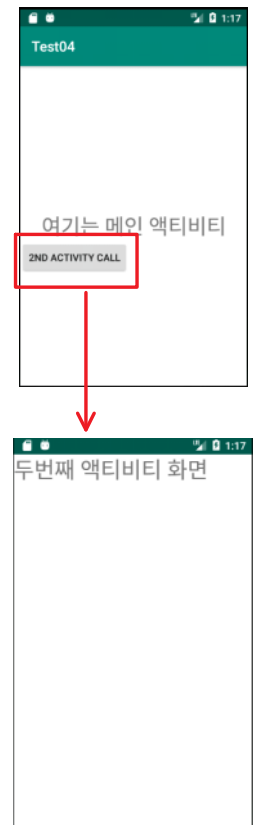
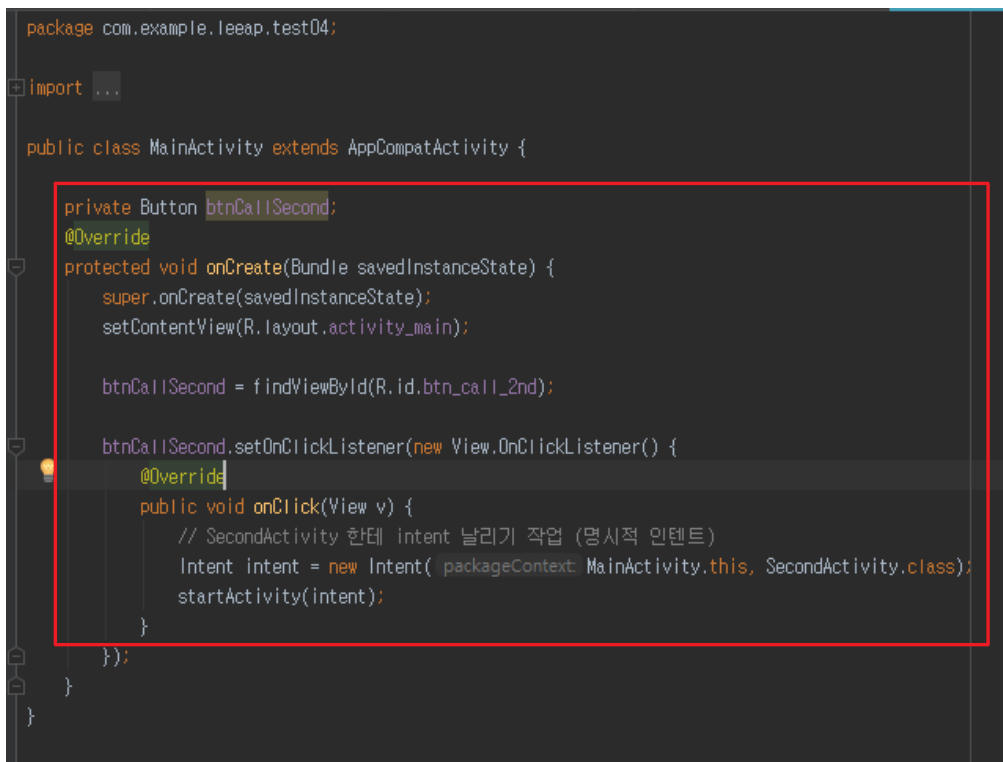
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //표지볼이기
        setContentView(R.layout.activity_second);
    }
}
```

AndroidManifest activity 추가



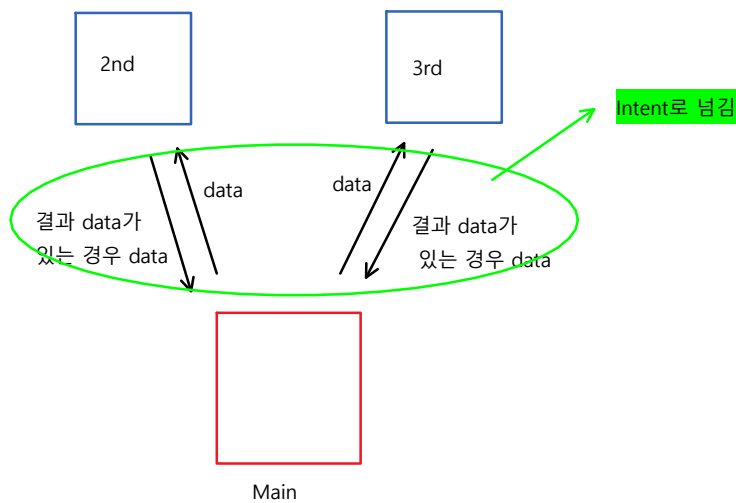
명시적 인텐트

//SecondActivity한테 intent 날리기 작업 > MainActivity.java



6. intent에 data보내기

2018년 11월 1일 목요일 오전 10:30



MainActivity 코드추가

```
btnCallSecond.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // SecondActivity 한데 intent 날리기 작업 (명시적 인텐트)
        Intent intent = new Intent( packageContext, MainActivity.this, SecondActivity.class);
        intent.putExtra( name: "msgTo2nd", value: "안녕 두번째 액티비티야~~?");
        startActivity(intent);
    }
});

btnCallThird.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View v) {
        Intent intent = new Intent( packageContext, MainActivity.this, ThirdActivity.class);
        intent.putExtra( name: "msgTo3rd", value: "안녕? 세번째 액티비티*~*");
        intent.putExtra( name: "now", new Date().toString());
        startActivity(intent);
    }
});
```

받을 변수 이름

보내는 데이터 값

SecondActivity 코드추가

```
public class SecondActivity extends Activity {
    private TextView textSecond;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //표지불이기
        setContentView(R.layout.activity_second);

        textSecond = findViewById(R.id.text_second);

        Intent receivedIntent = getIntent(); //MainActivity가 보낸 인텐트
        String receivedMsg = receivedIntent.getStringExtra( name: "msgTo2nd");
        textSecond.setText(textSecond.getText() + " //메인으로부터 받은 메세지 : " + receivedMsg);
    }
}
```

두번째 액티비티 화면//메
인으로부터 받은 메세지 :
안녕 두번째 액티비티야
~~?

ThirdActivity 코드추가

```

public class ThirdActivity extends Activity {

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

        Intent receivedIntent = getIntent(); //MainActivity가 보낸 인텐트
        String receivedMsg = receivedIntent.getStringExtra( name: "msgTo3nd");
        String receivedTime = receivedIntent.getStringExtra( name: "now");

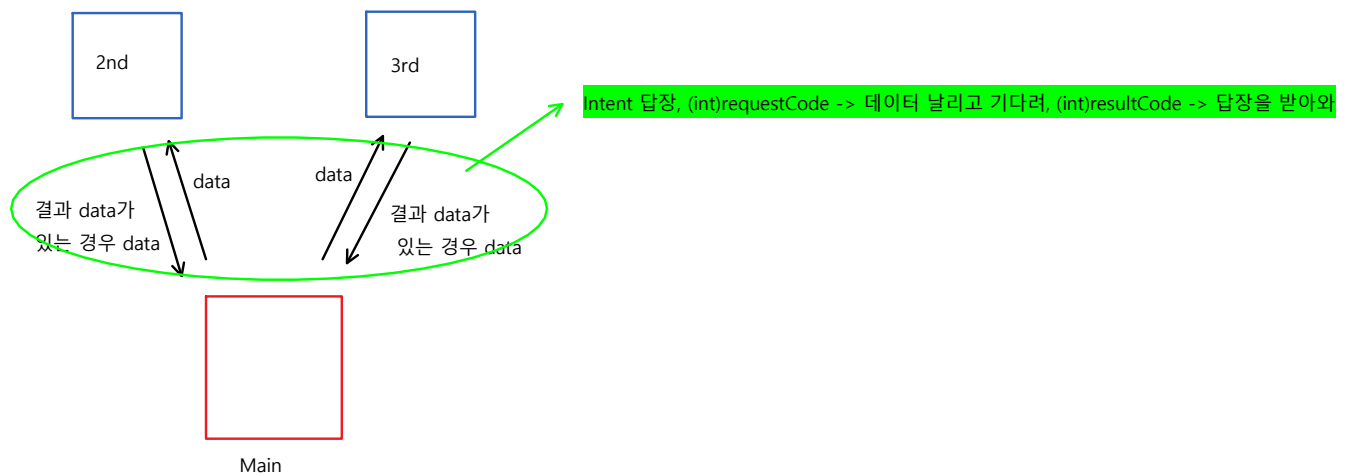
        Toast.makeText( context: this, text: receivedMsg+"//받은 시간: " + receivedTime, Toast.LENGTH_SHORT).show();
    }
}

```



7. Test05- 결과data보내기

2018년 11월 1일 목요일 오전 10:52



Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout 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"
    tools:context=".MainActivity">

    <WebView
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:id="@+id/webview"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toTopOf="@+id/btn_call"/>

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="목록 액티비티 호출"
        android:id="@+id/btn_call"
        app:layout_constraintBottom_toBottomOf="parent"/>

</android.support.constraint.ConstraintLayout>
```

MainActivity.java


```

public class MainActivity extends AppCompatActivity {

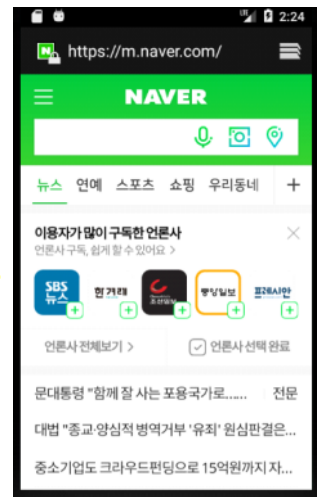
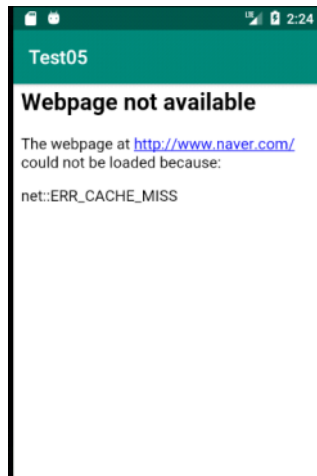
    private WebView webView;
    private Button btnCall;

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

        webView = findViewById(R.id.webview);
        btnCall = findViewById(R.id.btn_call);

        webView.loadUrl("http://www.naver.com");
    }
}

```



activity_main.xml × MainActivity.java × **AndroidManifest.xml**

```

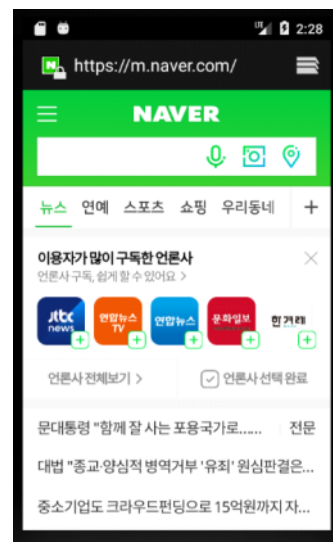
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.leeap.test05">
    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Test05"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="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>

```

앱에서 외부 인터넷 앱이 실행됨

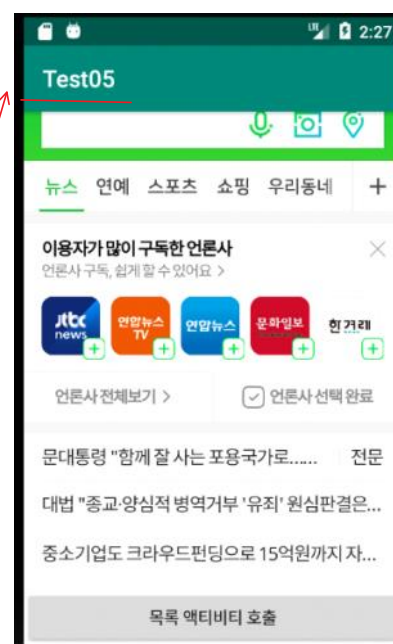


activity_main.xml × MainActivity.java × **AndroidManifest.xml**

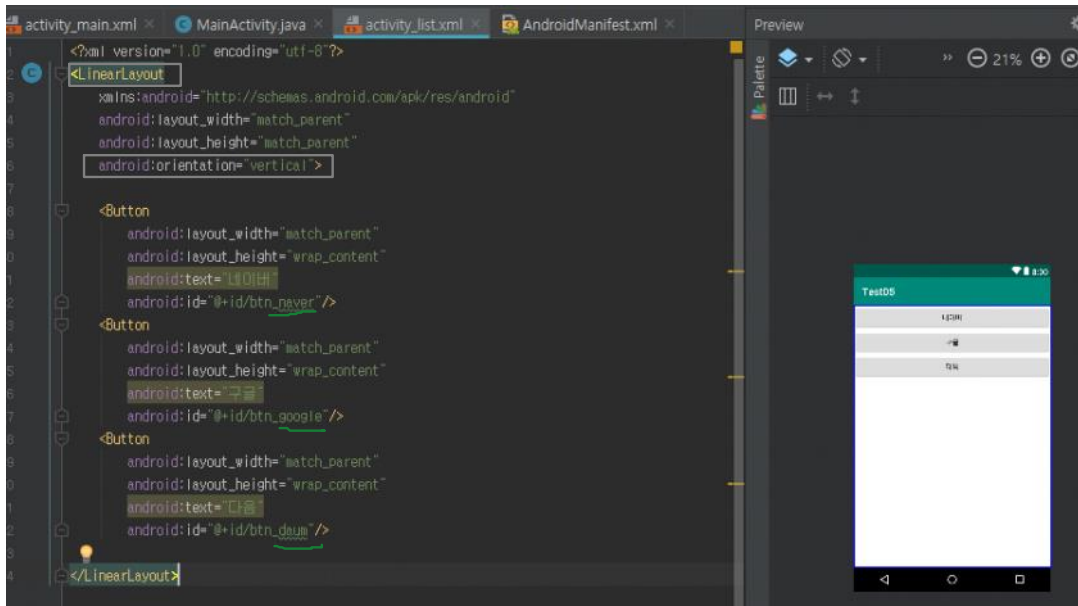
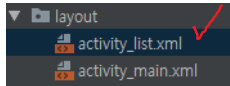
```

1 package com.example.leeap.test05;
2
3 import ...
4
5 public class MainActivity extends AppCompatActivity {
6
7     private WebView webView;
8     private Button btnCall;
9
10    @Override
11    protected void onCreate(Bundle savedInstanceState) {
12        super.onCreate(savedInstanceState);
13        setContentView(R.layout.activity_main);
14
15        webView = findViewById(R.id.webview);
16        btnCall = findViewById(R.id.btn_call);
17
18        //이거 안하면 앱 내부 브라우저가 아닌 외부 인터넷 앱이 실행됨 !!
19        webView.setWebViewClient(new WebViewClient());
20
21        webView.loadUrl("http://www.naver.com");
22    }
23
24 }
25
26
27
28
29

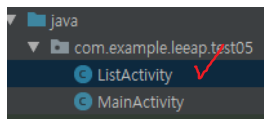
```



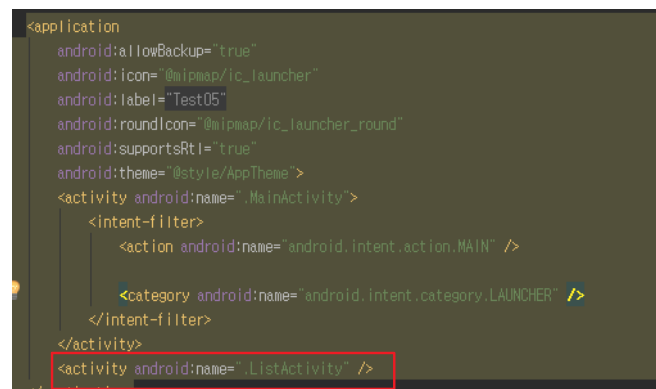
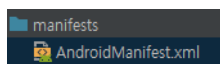
액티비티 추가

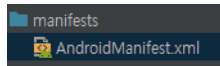


자바 클래스 추가



manifest 추가





```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="Test05"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportRtl="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>
    <activity android:name=".ListActivity" />
</application>
```

MainActivity

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

    webView = findViewById(R.id.webview);
    btnCall = findViewById(R.id.btn_call);

    webView.loadUrl("http://www.naver.com");

    btnCall.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent( packageContext MainActivity.this, ListActivity.class);

            //데이터 기다리는 코드
            startActivityForResult(intent, requestCode: 111);
        }
    });
}
```

MainActivity Intent data : 답장, (int)requestCode -> 데이터 날리고 기다려, (int)resultCode -> 답장을 받아와

```
public class ListActivity extends Activity {

    private Button btnNaver, btnGoogle, btnDaum;

    @Override
    protected void onCreate(Bundle savedInstanceState) {...}

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {

        if(requestCode == 111){
            if(requestCode == 200){ //사용자가 버튼을 선택한 경우
            }else if(requestCode == -1){ //사용자가 버튼 선택 X -> 그냥 back으로 돌아온상태
            }
        }
    }
}
```

```

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {

    if(requestCode == 111){ //내가 보내는 코드
        if(resultCode == 200){ //사용자가 버튼을 선택한 경우 받는 코드!
            String url = data.getStringExtra( "name: \"url\"");
            webView.loadUrl(url);
        }else if(resultCode == -1){ //사용자가 버튼 선택 X -> 그냥 back으로 돌아온상태
            Toast.makeText( context MainActivity.this, text: "back으로 돌아와서 url 변경이 없음!!", Toast.LENGTH_SHORT).show();
        }
    }
}

```

ListActivity

```

btnNaver.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View v){
        Intent intent = new Intent();
        intent.putExtra( "name: \"url\"", value: "http://www.naver.com");
        setResult(200,intent);
        finish();
    }
});

btnGoogle.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View v){
        Intent intent = new Intent();
        intent.putExtra( "name: \"url\"", value: "http://www.google.com");
        setResult( resultCode: 200,intent);
        finish();
    }
});

btnDaum.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View v){
        Intent intent = new Intent();
        intent.putExtra( "name: \"url\"", value: "http://www.daum.net");
        setResult( resultCode: 200,intent);
        finish();
    }
});

```

```

//백버튼 눌렀을 때
@Override
public void onBackPressed() {
    setResult(-1); //url 버튼 선택 안하고 main으로 돌아가는 resultCode
    finish();
}

```


8. Test06 - dialog

2018년 11월 2일 금요일 오전 9:20

Activity_main.xml

```
<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"
    tools:context=".MainActivity">

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/btn_basic"
        android:text="기본 다이얼로그" />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/btn_date"
        android:text="날짜 선택 다이얼로그" />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/btn_time"
        android:text="시간 선택 다이얼로그" />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/btn_custom"
        android:text="커스텀 다이얼로그" />

</LinearLayout>
```

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    private Button btnBasic, btnDate, btnTime, btnCustom;

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

        btnBasic = findViewById(R.id.btn_basic);
        btnDate = findViewById(R.id.btn_date);
        btnTime = findViewById(R.id.btn_time);
        btnCustom = findViewById(R.id.btn_custom);

        btnBasic.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
            }
        });

        btnDate.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
            }
        });

        btnTime.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
            }
        });

        btnCustom.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
            }
        });
    }
}
```

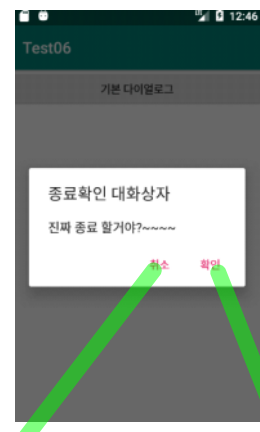
이벤트 처리 메소드를 새로 만들

Basic dialog 메소드

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    ...
}

public void showDialogBasic() {
    AlertDialog.Builder builder = new AlertDialog.Builder(context: this);

    Dialog dialog = builder.setTitle("종료확인 대화상자")
        .setMessage("진짜 종료 할거야?~~~~~")
        .setPositiveButton(text: "확인", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                //확인 버튼 누르면 -> 앱 종료
                MainActivity.this.finish();
            }
        })
        .setNegativeButton(text: "취소", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
            }
        })
        .create();
    dialog.show();
}
```



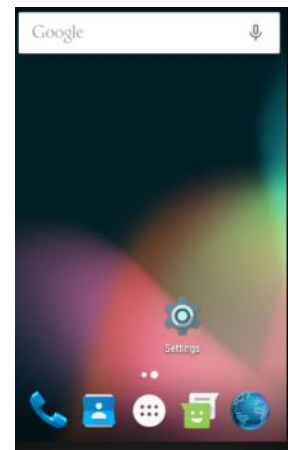
```

        //확인 버튼 누르면 -> 앱 종료
        MainActivity.this.finish();
    }
}

.setNegativeButton( text: "취소", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        //취소 버튼 누르면 -> 대화상자만 사라지게
        dialog.cancel();
    }
}).create();

dialog.show();
}

```



호출

```

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

    btnBasic = findViewById(R.id.btn_basic);
    btnDate = findViewById(R.id.btn_date);
    btnTime = findViewById(R.id.btn_time);
    btnCustom = findViewById(R.id.btn_custom);

    btnBasic.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            showDialogBasic();
        }
    });
}

```

MainActivity.java 날짜선택 daialog 메소드

```

public class MainActivity extends AppCompatActivity {
    private Button btnBasic, btnDate, btnTime, btnCustom;

    @Override
    protected void onCreate(Bundle savedInstanceState) {...}

    public void showDialogBasic(){...}

    public void showDialogDate(){
        int dYear, dMonth, dDay; //기본 날짜 세팅

        Calendar c = Calendar.getInstance();

        dYear = c.get(Calendar.YEAR);
        dMonth = c.get(Calendar.MONTH);
        dDay = c.get(Calendar.DAY_OF_MONTH);

        DatePickerDialog.OnDateSetListener listener = new DatePickerDialog.OnDateSetListener() {
            @Override
            public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {
                Toast.makeText( context: MainActivity.this,
                    text: "선택한 날짜 "+year+"년 "+(month+1)+"월 "+dayOfMonth+"일",
                    Toast.LENGTH_SHORT).show();
            }
        };

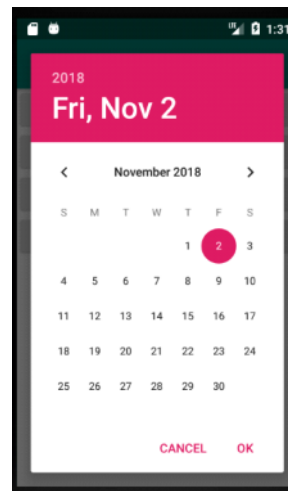
        // 날짜선택, 선택시처리, 년, 월, 일 기본값
        DatePickerDialog dateDialog = new DatePickerDialog( context: this, listener, dYear, dMonth, dDay);
        dateDialog.show();
    }
}

```

```

btnDate.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        showDialogDate();
    }
});

```



MainActivity.java 시간선택 daialog 메소드

```

public void showDialogTime(){
    int dHour, dMin;

    Calendar c = Calendar.getInstance();

    dHour = c.get(Calendar.HOUR_OF_DAY);
    dMin = c.get(Calendar.MINUTE);

    TimePickerDialog.OnTimeSetListener listener = new TimePickerDialog.OnTimeSetListener() {
        @Override
        public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
            Toast.makeText(context, MainActivity.this,
                text: "선택한 시간 : " + hourOfDay + "시" + minute + "분",
                Toast.LENGTH_SHORT).show();
        }
    };

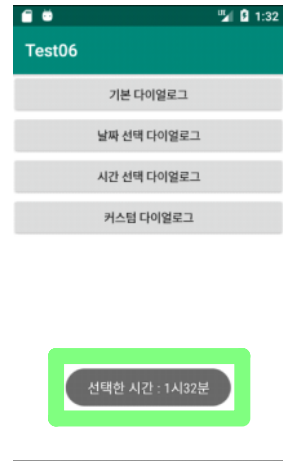
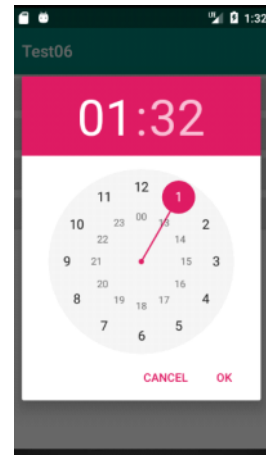
    TimePickerDialog timeDialog = new TimePickerDialog(context, this, listener, dHour, dMin, is24HourView, true);
    timeDialog.show();
}

```

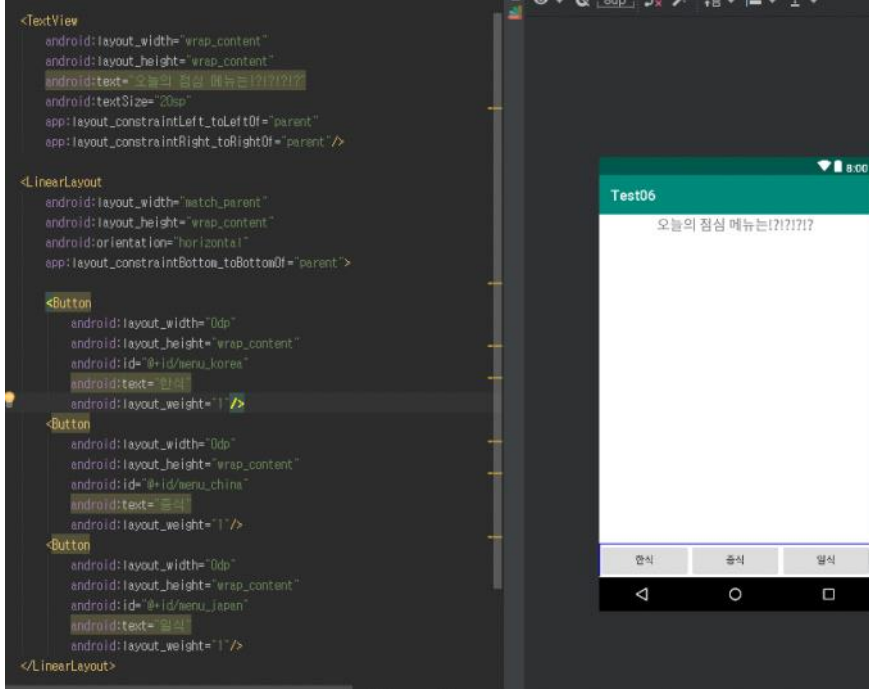
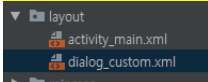
```

btnTime.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        showDialogTime();
    }
});

```



커스텀 dialog 메소드 Dialog_custom.xml만들기



MainActivity.java 커스텀 dialog 메소드


```

public void showDialogCustom(){
    Dialog dialog = new Dialog( context: this);
    dialog.setContentView(R.layout.dialog_custom); //inflate

    Button btnKorea, btnChina, btnJapan;
    btnKorea = dialog.findViewById(R.id.menu_korea);
    btnChina = dialog.findViewById(R.id.menu_china);
    btnJapan = dialog.findViewById(R.id.menu_japan);

    btnKorea.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Toast.makeText( context: MainActivity.this, text: "한식중에 뭐 먹지..", Toast.LENGTH_SHORT).show();
        }
    });

    btnChina.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Toast.makeText( context: MainActivity.this, text: "난 짬뽕 !..", Toast.LENGTH_SHORT).show();
        }
    });

    btnJapan.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Toast.makeText( context: MainActivity.this, text: "소바소바", Toast.LENGTH_SHORT).show();
        }
    });

    dialog.show();
}

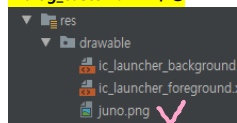
```

```

btnCustom.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        showDialogCustom();
    }
});

```

Dialog_custom.xml 수정

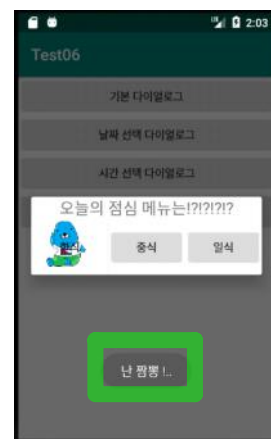
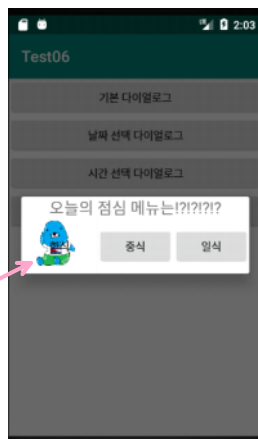


```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="오늘의 점심 메뉴는!?!?!?"
    android:textSize="20sp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintBottom_toBottomOf="@+id/group_btns"
    android:id="@+id/text_menu"/>

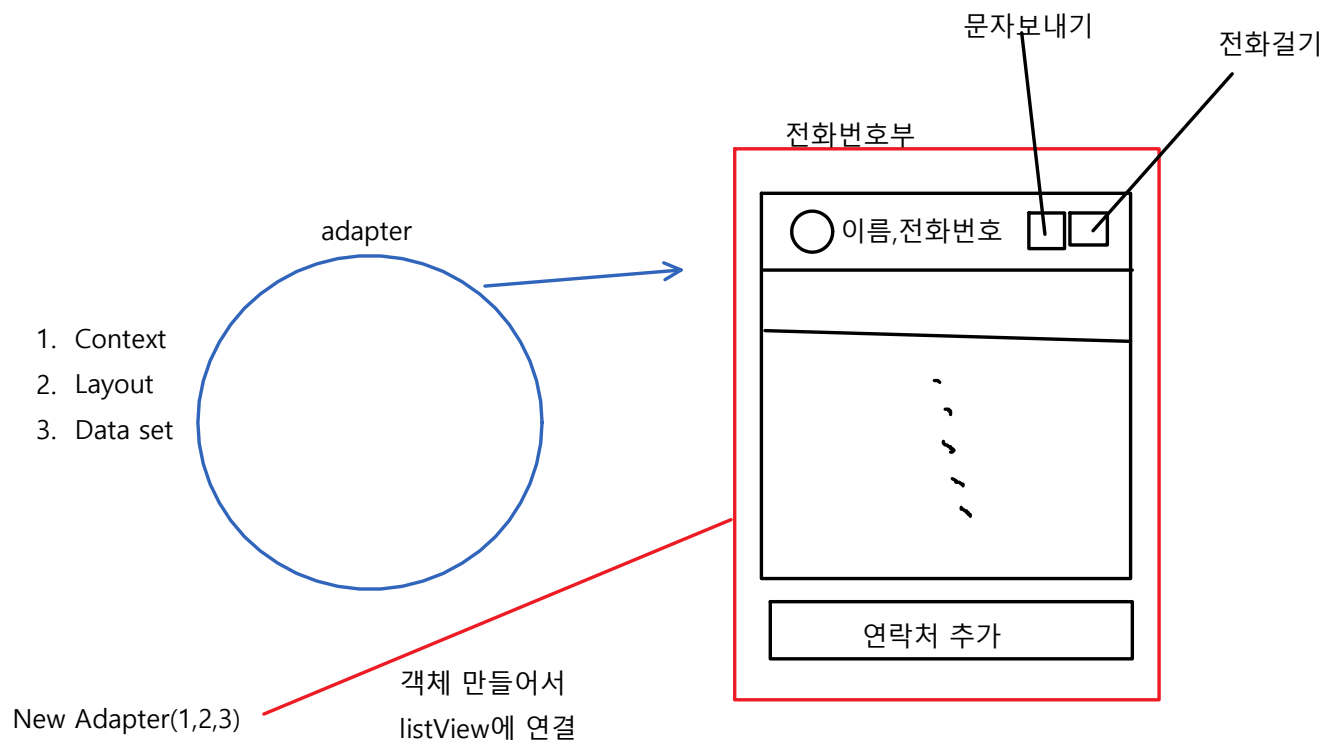
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:id="@+id/group_btns"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintTop_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent">
    <Button
        android:layout_width="0dp"
        android:layout_height="70dp"
        android:id="@+id/menu_korea"
        android:text="한식"
        android:layout_weight="1"
        android:background="@drawable/juno"/>
    <Button
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:id="@+id/menu_japan"
        android:text="일식"
        android:layout_weight="1"/>
</LinearLayout>

```



9. Test07-전화번호부

2018년 11월 5일 월요일 오전 9:43



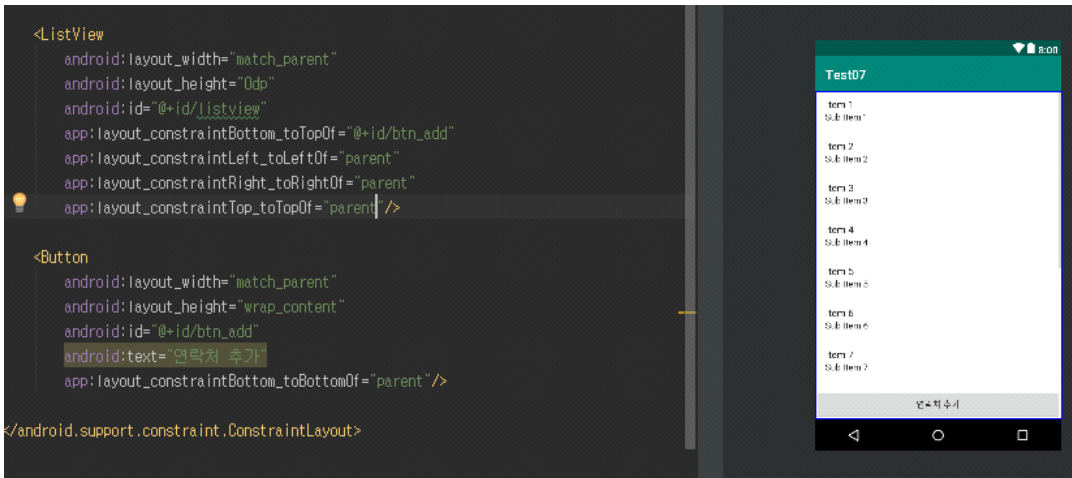
순서 -

1. MainActivity 화면 레이아웃, xml
2. 리스트 뷰의 아이템 레이아웃 - xml
3. ContactsAdapter 클래스 작성(context, layout, dataSet)
4. MainActivity가 가지고 있는 ListView한테 어댑터 객체 등록

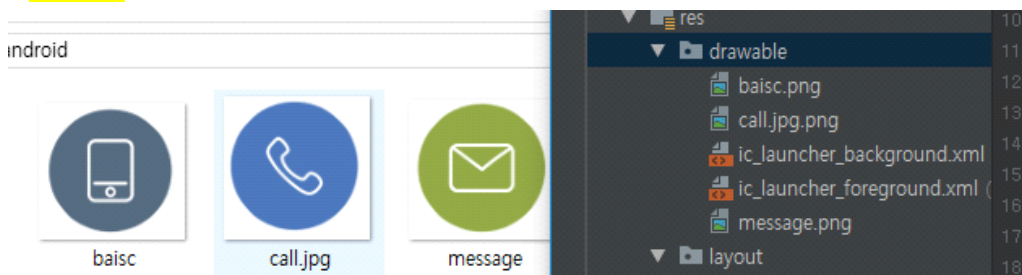
전화번호부 리스트

2018년 11월 5일 월요일 오전 10:05

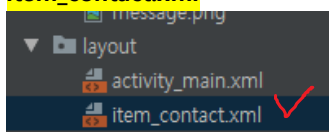
activity_main.xml



사진추가



item_contact.xml



```

<ImageView
    android:layout_width="60dp"
    android:layout_height="60dp"
    android:id="@+id/contact_img"
    android:background="@drawable/baisc"
    android:layout_marginLeft="10dp"
    app:layout_constraintLeft_toLeftOf="parent"/>

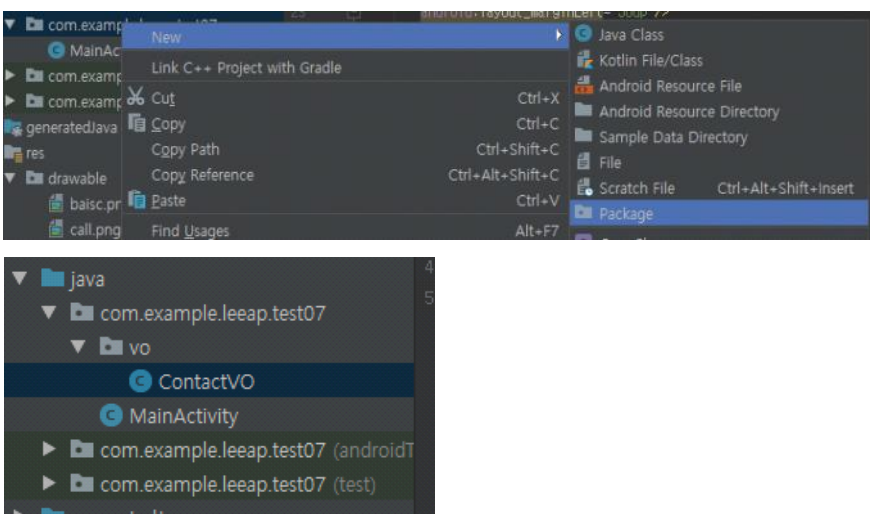
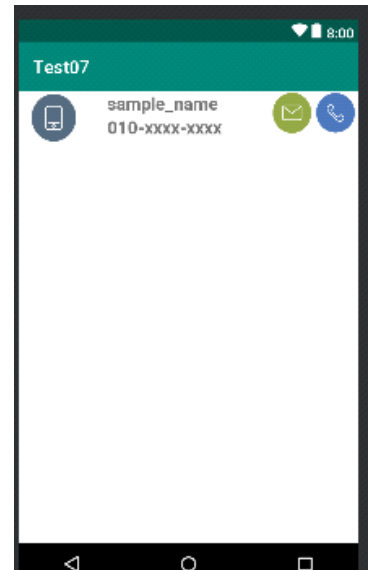
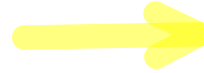
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="sample_name"
    android:textStyle="bold"
    android:id="@+id/contact_name"
    app:layout_constraintLeft_toRightOf="@id/contact_img"
    android:textSize="20sp"
    android:layout_marginLeft="30dp"/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="010-xxxx-xxxx"
    android:id="@+id/contact_phone"
    app:layout_constraintLeft_toRightOf="@id/contact_img"
    app:layout_constraintTop_toBottomOf="@id/contact_name"
    android:textSize="20sp"
    android:textStyle="bold"
    android:layout_marginLeft="30dp"/>

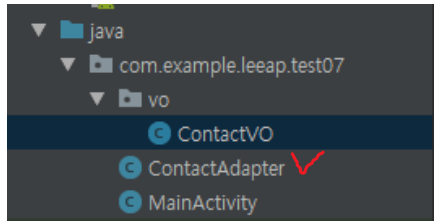
<Button
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:background="@drawable/call"
    app:layout_constraintRight_toRightOf="parent"
    android:id="@+id/contact_btn_call"/>

<Button
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:background="@drawable/message"
    app:layout_constraintRight_toLeftOf="@+id/contact_btn_call"
    android:id="@+id/contact_btn_sms"/>

```

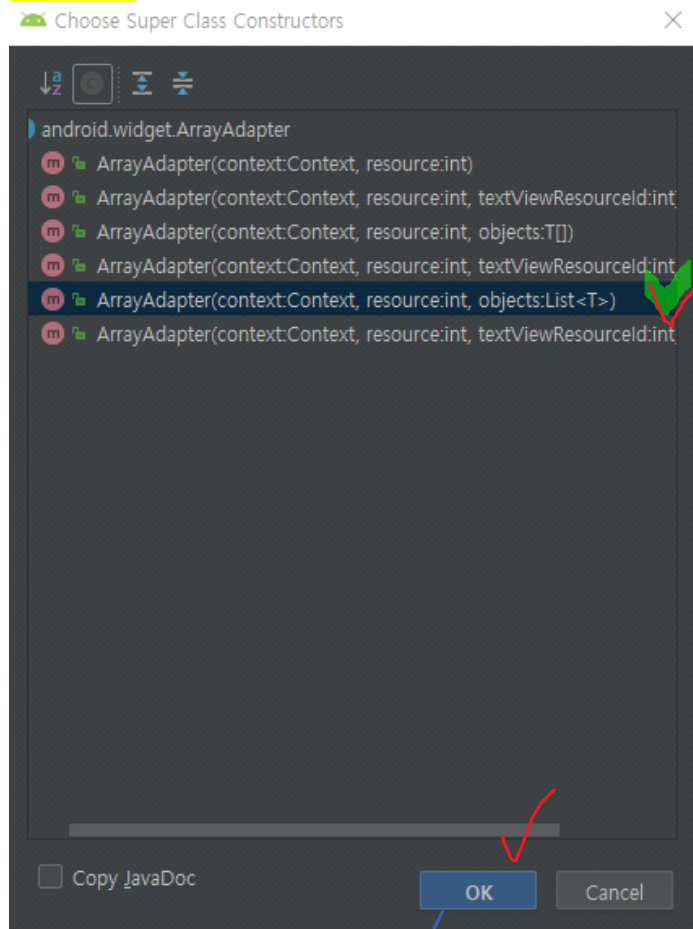


contactAdapter.java



```
public class ContactAdapter extends ArrayAdapter<ContactVO> {
```

Alt + enter



```
public class ContactAdapter extends ArrayAdapter<ContactVO> {  
  
    public ContactAdapter(Context context, int resource, List<ContactVO> objects) {  
        super(context, resource, objects);  
    }  
}
```

```

public class ContactAdapter extends ArrayAdapter<ContactVO> {

    private Context context;
    private int itemLayout;
    private List<ContactVO> contactList;

    public ContactAdapter(Context context, int resource, List<ContactVO> objects) {
        super(context, resource, objects);
        this.context = context;
        this.itemLayout = resource;
        this.contactList = objects;
    }
}

```

MainActivity.java

```

public class MainActivity extends AppCompatActivity {

    private ListView listView;
    private Button btnAdd;
    private ContactAdapter adapter;

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

        listView = findViewById(R.id.listView);
        btnAdd = findViewById(R.id.btn_add);

        //객체생성
        adapter = new ContactAdapter( context: this, R.layout.item_contact, new ArrayList<ContactVO>());

        //장착
        listView.setAdapter(adapter);
    }
}

```

ContactAdapter.java

```

@Override
public View getView(int position, View convertView, ViewGroup parent){
    return super.getView(position, convertView, parent);
}

```

```

@Override //자동으로 위치지정
public View getView(int position, View convertView, ViewGroup parent){
    LayoutInflater inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    View item = inflater.inflate(itemLayout, parent); //R.layout.item_contact

    TextView tvContactName = item.findViewById(R.id.contact_name);
    TextView tvContactPhone = item.findViewById(R.id.contact_phone);
    Button contactBtnCall = item.findViewById(R.id.contact_btn_call);
    Button contactBtnSms = item.findViewById(R.id.contact_btn_sms);

    return super.getView(position, convertView, parent);
}

```

MainActivity.java

```

public class MainActivity extends AppCompatActivity {

    private ListView listView;
    private Button btnAdd;
    private ContactAdapter adapter;

    private ArrayList<ContactVO> contactList;

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

        listView = findViewById(R.id.listview);
        btnAdd = findViewById(R.id.btn_add);

        contactList = new ArrayList<ContactVO>();

        for(int i = 0; i<20; i++){
            contactList.add(new ContactVO( name: "정은"+i, phoneNum: "01092743036"));
        }

        //객체생성
        adapter = new ContactAdapter( context: this, R.layout.item_contact, contactList);

        //장착
        listView.setAdapter(adapter);
    }
}

```

ContactAdapter.java

```

@Override //자동으로 위치지정
public View getView(int position, View convertView, ViewGroup parent){
    LayoutInflater inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    View item = inflater.inflate(itemLayout, root, null); layout.item_contact

    TextView tvContactName = item.findViewById(R.id.contact_name);
    TextView tvContactPhone = item.findViewById(R.id.contact_phone);
    Button contactBtnCall = item.findViewById(R.id.contact_btn_call);
    Button contactBtnSms = item.findViewById(R.id.contact_btn_sms);

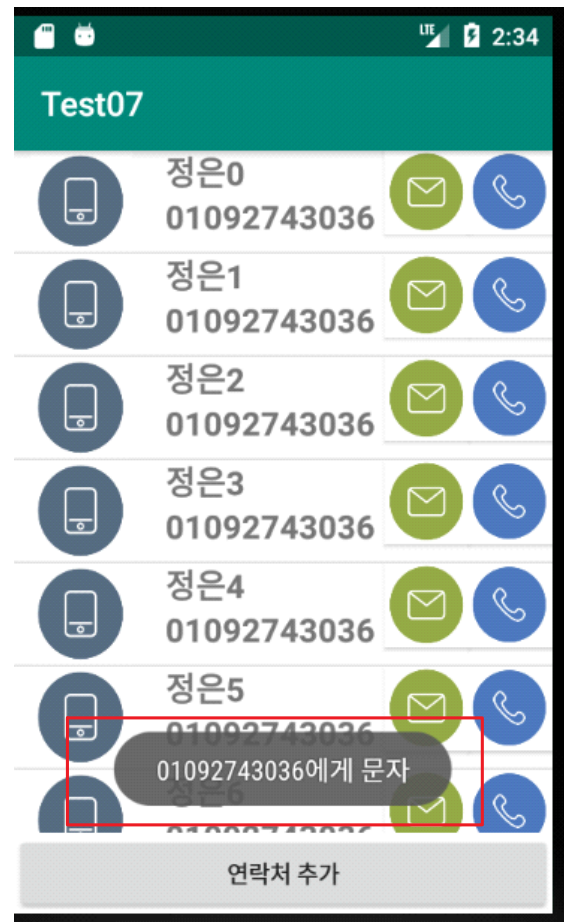
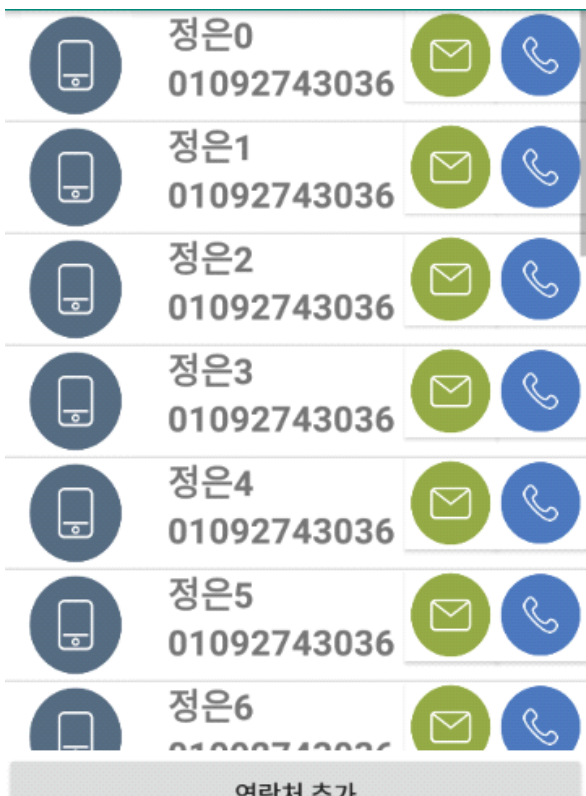
    //데이터 집합에서 연락처 객체 하나 꺼내기
    final ContactVO contactVO = contactList.get(position);

    tvContactName.setText(contactVO.getName());
    tvContactPhone.setText(contactVO.getPhoneNum());

    contactBtnCall.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Toast.makeText(context, text: contactVO.getPhoneNum()+"에게 전화", Toast.LENGTH_SHORT).show();
        }
    });
    contactBtnSms.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Toast.makeText(context, text: contactVO.getPhoneNum()+"에게 문자", Toast.LENGTH_SHORT).show();
        }
    });

    return item;
}

```



Intent - 전화, 문자 기능

2018년 11월 5일 월요일 오전 11:33

ContactAdapter.java

```
contactBtnCall.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Toast.makeText(context, "text: " + contactVO.getPhoneNum() + "에게 전화", Toast.LENGTH_SHORT).show();
    }
});
contactBtnSms.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Toast.makeText(context, "text: " + contactVO.getPhoneNum() + "에게 문자", Toast.LENGTH_SHORT).show();
    }
});
```

전화기능 추가

문자기능 추가

```
contactBtnCall.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent();
        intent.setAction(Intent.ACTION_CALL);
        intent.setData(Uri.parse("tel:" + contactVO.getPhoneNum()));
        context.startActivity(intent);
    }
});
```

```
contactBtnSms.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Toast.makeText(context, contactVO.getPhoneNum() + "에게 문자", Toast.LENGTH_SHORT).show();
        Intent intent = new Intent();
        intent.setAction(Intent.ACTION_SENDTO);
        intent.setData(Uri.parse("sms:" + contactVO.getPhoneNum()));
        intent.putExtra("sms_body", "이 내용으로 문자 보내");
        context.startActivity(intent);
    }
});
```

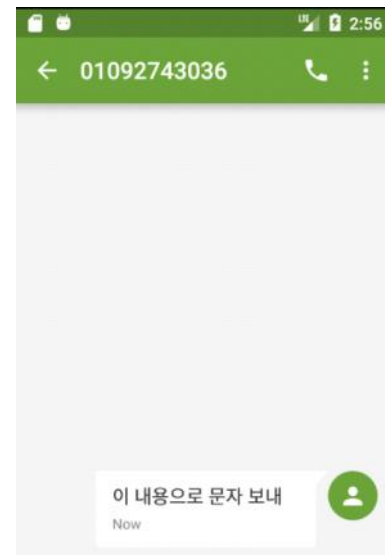
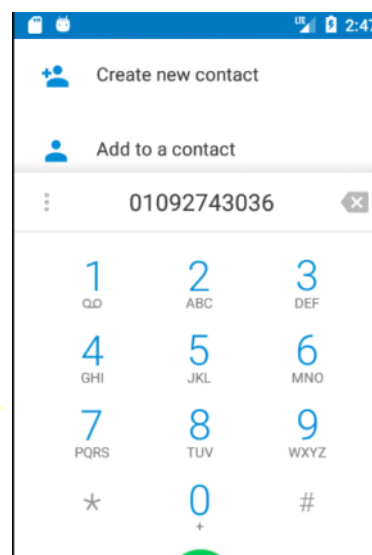
AndroidManifest.xml

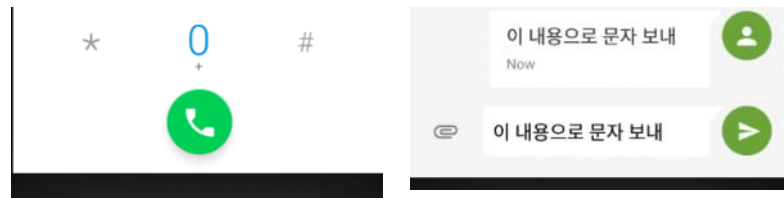
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.leeap.test07">
    <uses-permission android:name="android.permission.CALL_PHONE"/>
    <uses-permission android:name="android.permission.SEND_SMS"/>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Test07"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="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>
```

전화, 문자 기능 추가

```
contactBtnCall.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent();
        intent.setAction(Intent.ACTION_DIAL);
        intent.setData(Uri.parse("tel:" + contactVO.getPhoneNum()));
        context.startActivity(intent);
    }
});
```

전화 걸기는 사용자의 동의를 받아야 해서
일단 다이얼로 Test

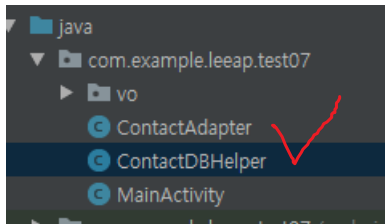




DB연결

2018년 11월 6일 화요일 오전 9:14

ContactDBHelper.java



public class ContactDBHelper extends SQLiteOpenHelper {
 @Override

this
버전 올리는 작업등
그냥 이름 지어주면 됨

```
public class ContactDBHelper extends SQLiteOpenHelper {  
    public ContactDBHelper( Context context, String name, SQLiteDatabase.CursorFactory factory, int version) {  
        super(context, name, factory, version);  
    }  
  
    @Override  
    public void onCreate(SQLiteDatabase db) {  
    }  
  
    @Override  
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
    }  
}
```

```
public class ContactDBHelper extends SQLiteOpenHelper {  
    private static final String DB_NAME = "my_contact.db";  
    private static final int DB_VERSION = 1;  
    private SQLiteDatabase db;  
  
    public ContactDBHelper( Context context, String name, SQLiteDatabase.CursorFactory factory, int version) {  
        super(context, name, factory, null, version);  
    }  
  
    //db = getReadableDatabase(); // 사용법, 동작이 둘다 똑같은  
    db = getWritableDatabase(); //데이터 베이스 연결 생성  
}  
  
@Override  
public void onCreate(SQLiteDatabase db) {  
    // my_contact.db파일이 처음 생성될때 1번 실행되는 메소드  
}  
  
@Override  
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
    // version 정보가 바뀔 시점에 실행되는 메소드 (version 숫자는 커지는 것만 가능함)  
}  
}
```

Table 생성 sql

Table 삭제후 다시 생성 sql

Table 생성 sql

```
@Override
public void onCreate(SQLiteDatabase db) {
    // my_contact.db파일이 처음 생성될때 1번 실행되는 메소드
    String sql = "CREATE TABLE CONTACT_TAB(" +
        "CONTACT_NUM INTEGER PRIMARY KEY AUTOINCREMENT," +
        "CONTACT_NAME TEXT," +
        "CONTACT_PHONE TEXT )";
    db.execSQL(sql);
}
```

Table 삭제후 다시 생성 sql

```
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    // version 정보가 바뀐 시점에 실행되는 메소드 (version 숫자는 커지는 것만 가능함)
    // 이번 예제에서는 버전이 바뀌면 테이블을 삭제하고 다시 재생성 하게 할거임
    String sql = "DROP TABLE IF EXISTS CONTACT_TAB";
    db.execSQL(sql);
    //테이블 다시 create
    onCreate(db);
}
```

CRUD 작업

INSERT

```
public void insert(ContactVO contactVO){
    String sql = "INSERT INTO CONTACT_TAB(CONTACT_NAME, CONTACT_PHONE)" +
        "VALUES(" + contactVO.getName() + "," + contactVO.getPhoneNum() + ")";
    db.execSQL(sql);
}
```

```
public void insert(ContactVO contactVO){
    ContentValues values = new ContentValues();
    values.put("CONTACT_NAME", contactVO.getName());
    values.put("CONTACT_PHONE", contactVO.getPhoneNum());
    db.insert( table: "CONTACT_TAB", nullColumnHack: null, values);
}
```

SELECTALL

```
public ArrayList<ContactVO> selectAll(){
    // 혹시 where문 쓸때는 이렇게
    // String sql = "SELECT CONTACT_NUM, CONTACT_NAME, CONTACT_PHONE FROM CONTACT_TAB" +
    // "WHERE CONTACT_NUM > ? AND CONTACT <?";
    // Cursor cursor = db.rawQuery(sql, new String[]{"10", "20"});

    String sql = "SELECT CONTACT_NUM, CONTACT_NAME, CONTACT_PHONE FROM CONTACT_TAB";
    Cursor cursor = db.rawQuery(sql, selectionArgs: null);

    ArrayList<ContactVO> contactList = new ArrayList<>();

    //r 같은거 , 다음게 있으면 true 없으면 false
    while (cursor.moveToNext()){
        int contactNum = cursor.getInt( columnIndex: 0); //0번컬럼 고유번호
        String contactName = cursor.getString( columnIndex: 1); //1번컬럼 이름
        String contactPhone = cursor.getString( columnIndex: 2); //2번컬럼 번호

        contactList.add(new ContactVO(contactName,contactPhone));
    }
    return contactList;
}
```

ContactDBHelper.java 수정

```
public class ContactDBHelper extends SQLiteOpenHelper {
    private static final String DB_NAME = "my_contact.db";
    private static final int DB_VERSION = 1;

    private SQLiteDatabase db;

    public ContactDBHelper( Context context) {
        super(context, DB_NAME, null, DB_VERSION);
        //db = getReadableDatabase(); // 사용법, 동작이 둘다 똑같은
        db = getWritableDatabase(); //데이터 베이스 연결 생성
    }
}
```

MainActivity.java 수정

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

    listView = findViewById(R.id.listView);
    btnAdd = findViewById(R.id.btn_add);

    //
    //     contactList = new ArrayList<ContactVO>();
    //
    //     for(int i = 0; i<20; i++){
    //         contactList.add(new ContactVO("정은"+i, "01092743036"));
    //     }

    ContactDBHelper dbHelper = new ContactDBHelper( context this);
    contactList = dbHelper.selectAll();

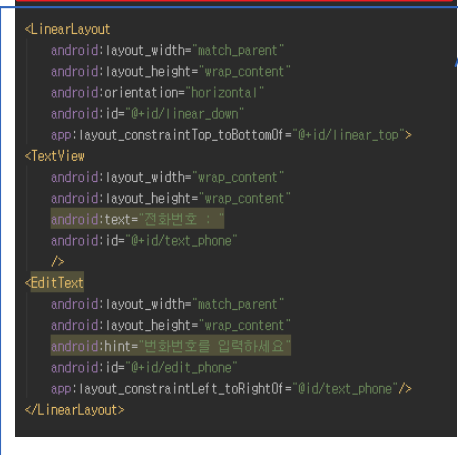
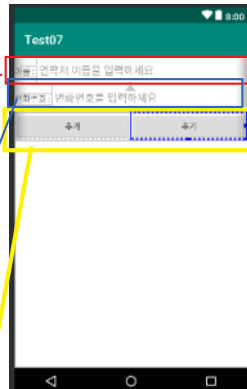
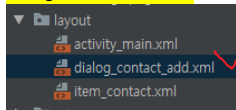
    //객체생성
    adapter = new ContactAdapter( context this,R.layout.item_contact, contactList);

    //장착
    listView.setAdapter(adapter);
}
}
```

연락처 추가

2018년 11월 6일 화요일 오전 10:22

dialog_contact_add.xml



MainActivity.java -> 전화번호 추가 layout 나타내기

```
private ContactDBHelper dbHelper = new ContactDBHelper( context, this);
```

지역변수로 빼기

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

    listView = findViewById(R.id.listView);
    btnAdd = findViewById(R.id.btnAdd);

    contactList = new ArrayList<ContactVO>();

    for(int i = 0; i<20; i++){
        contactList.add(new ContactVO("정은"+i, "01092743036"));
    }

    contactList = dbHelper.selectAll();
    //객체생성
    adapter = new ContactAdapter( context: this,R.layout.item_contact, contactList);
    //장착
    listView.setAdapter(adapter);

    btnAdd.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            showContactAddDialog();
        }
    });
}
```

```

public void showContactAddDialog(){
    final Dialog dialog = new Dialog( context: this);
    dialog setContentView(R.layout.dialog_contact_add);

    final EditText editName = dialog.findViewById(R.id.edit_name);
    final EditText editPhone = dialog.findViewById(R.id.edit_phone);
    Button btnDialogAdd = dialog.findViewById(R.id.btn_dialog_add);
    Button btnDialogCancel = dialog.findViewById(R.id.btn_dialog_cancel);

    btnDialogAdd.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            //다이얼로그에서 추가 버튼을 클릭한 경우 DB에 insert시켜야함
            ContactVO contactVO = new ContactVO();
            contactVO.setName(editName.getText()+"");
            contactVO.setPhoneNum(editPhone.getText()+"");

            dbHelper.insert(contactVO); // db에 추가하고 다이얼로그 닫기
            dialog.cancel();
        }
    });

    btnDialogCancel.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            dialog.cancel(); // 취소 버튼 누르면 그냥 다이얼로그 끄기
        }
    });

    dialog.show();
}

```

예러 - MainActivity.java

```

private ArrayList<ContactVO> contactList;
private ContactDBHelper dbHelper;

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

    listView = findViewById(R.id.listView);
    btnAdd = findViewById(R.id.btnAdd);

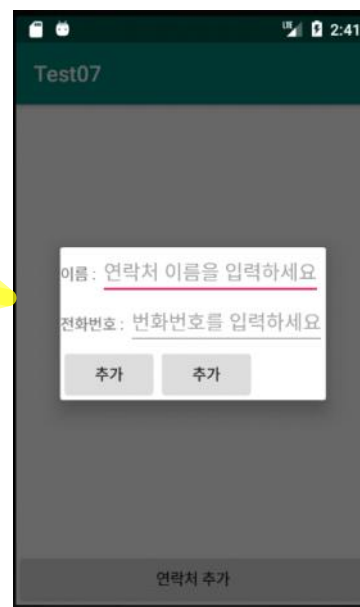
    contactList = new ArrayList<ContactVO>();

    for(int i = 0; i<20; i++){
        contactList.add(new ContactVO("장은"+i, "01092743036"));
    }

    dbHelper = new ContactDBHelper( context: this);
    contactList = dbHelper.selectAll();
    //적체생성
    adapter = new ContactAdapter( context: this,R.layout.item_contact, contactList);
    //장착
    listView.setAdapter(adapter);

    btnAdd.setOnClickListener(new View.OnClickListener() {

```



예러2 - MainActivity.java

```

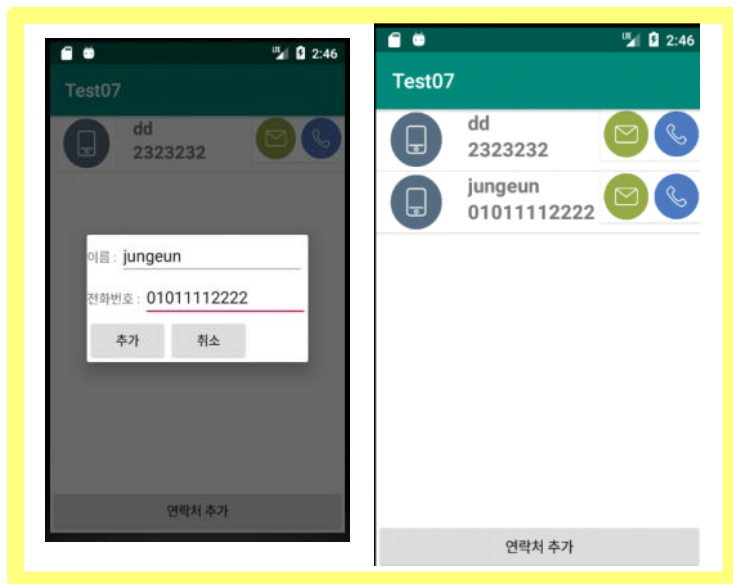
@Override
public void onClick(View v) {
    //다이얼로그에서 추가 버튼을 클릭한 경우 DB에 insert시켜야함
    ContactVO contactVO = new ContactVO();
    contactVO.setName(editName.getText()+"");
    contactVO.setPhoneNum(editPhone.getText()+"");

    dbHelper.insert(contactVO); // db에 추가하고 다이얼로그 닫기

    // 받금 db에 변경이 일어났으니, listView쪽도 갱신되게 해줘야 바로 반영된것이 확인됨
    contactList.clear();
    ArrayList<ContactVO> tmpList = dbHelper.selectAll();
    contactList.addAll(tmpList);
    adapter.notifyDataSetChanged(); //listview 담당 어댑터에게 데이터 변경 알리기!!!
    dialog.cancel();
}
}

```





10. Test08 - 스레드

2018년 11월 8일 목요일 오전 9:13

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="카운트!"
    android:textSize="50dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    android:id="@+id/tv_count"/>
```

MainActivity.java

```
package com.example.leeap.test08;

import ...

public class MainActivity extends AppCompatActivity {
    private TextView tvCount;

    private int count = 100;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        tvCount = findViewById(R.id.tv_count);

        new CountThread().start();
    }

    //Thread로써의 동작이 필요한 요소들이 포함된 클래스 상속
    class CountThread extends Thread{
        @Override
        public void run() {
            for(count=100; count>=0; count--) {
                tvCount.setText(count + "");
                try {
                    Thread.sleep(1000);
                } catch (Exception e){
                    e.printStackTrace();
                }
            }
        }
    }
}
```

Error!

스레드가 직접적으로 view를 건드리면 안돼

메인 스레드 -> 화면 그리는 작업 전담해야 함

CountThread.java 별도 클래스로 작성

```
com.example.leeap.test08
├── CountThread ✓
└── MainActivity
```

```
package com.example.leeap.test08;

import android.os.Handler;
import android.os.Message;
```

MainActivity.java

```
package com.example.leeap.test08;

import ...

public class MainActivity extends AppCompatActivity {
```

```

package com.example.test08;

import android.os.Handler;
import android.os.Message;

public class CountThread extends Thread{
    private Handler mainHandler;

    public CountThread(Handler handler){
        this.mainHandler = handler;
    }

    @Override
    public void run() {
        for(int count=100; count>=0; count--){
            //tvCount.setText(count + "");
            Message sendMsg = new Message();
            sendMsg.what = count;
            mainHandler.sendMessage(sendMsg);

            try {
                Thread.sleep(1000);
            } catch (Exception e){
                e.printStackTrace();
            }
        }
    }
}

```

```

import ...

public class MainActivity extends AppCompatActivity {
    private TextView tvCount;
    //비서역할 무명클래스
    private Handler myHandler = new Handler(){

        @Override
        public void handleMessage(Message msg) {
            // 다른 스레드에서 메세지 객체가 왔을 때 호출되는 메소드
            int receiveCount = msg.what;
            tvCount.setText(receiveCount+"");
        }
    };

    private int count = 100;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

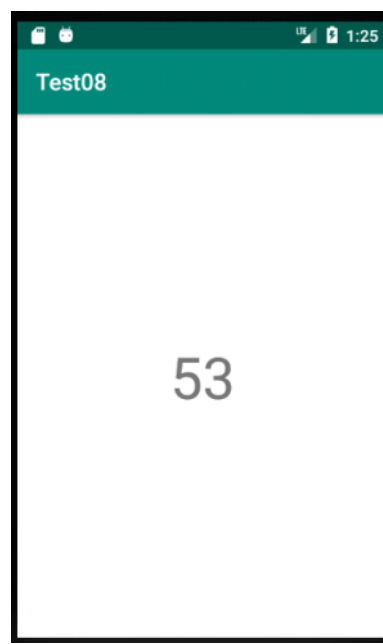
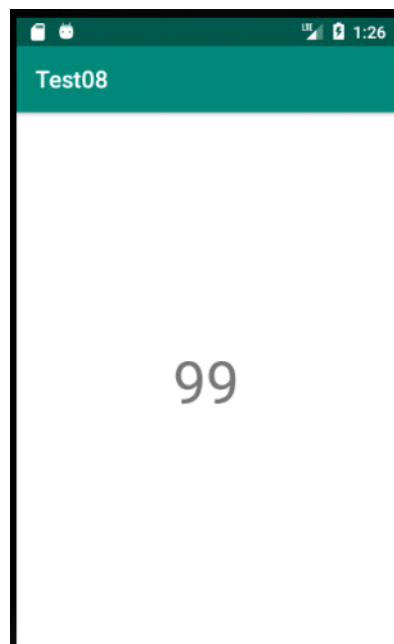
        tvCount = findViewById(R.id.tv_count);

        new CountThread(myHandler).start();
    }
}

```



1초마다 count down



11. Test09

2018년 11월 8일 목요일 오전 10:12

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    private TextView tvCount;

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

        tvCount = findViewById(R.id.tv_count);

        new CountTask().execute();
    }

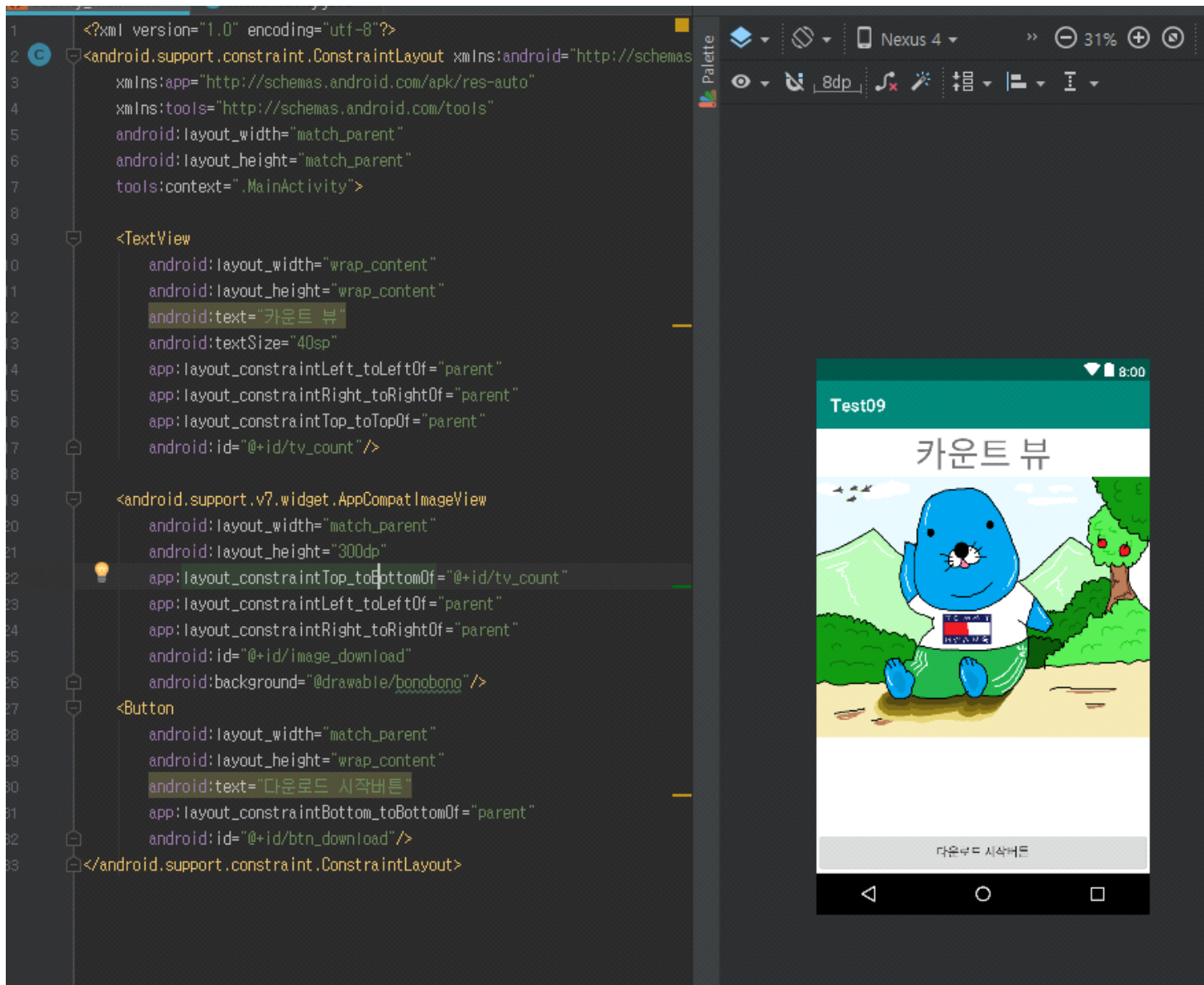
    class CountTask extends AsyncTask<Void, Void, Void> {
        // @Override
        // protected void onPreExecute() {
        //     super.onPreExecute();
        // }
        private int count;

        @Override
        protected Void doInBackground(Void... voids) {
            for (count = 100; count >= 0; count--) {
                // tvCount.setText(count + "");
                publishProgress();
                try {
                    Thread.sleep(1000);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }

            tvCount.setText("Complete!!");
            return null;
        }

        @Override
        protected void onProgressUpdate(Void... values) {
            tvCount.setText(count + "");
        }

        @Override
        protected void onPostExecute(Void aVoid) {
            tvCount.setText("Complete!");
        }
    }
}
```



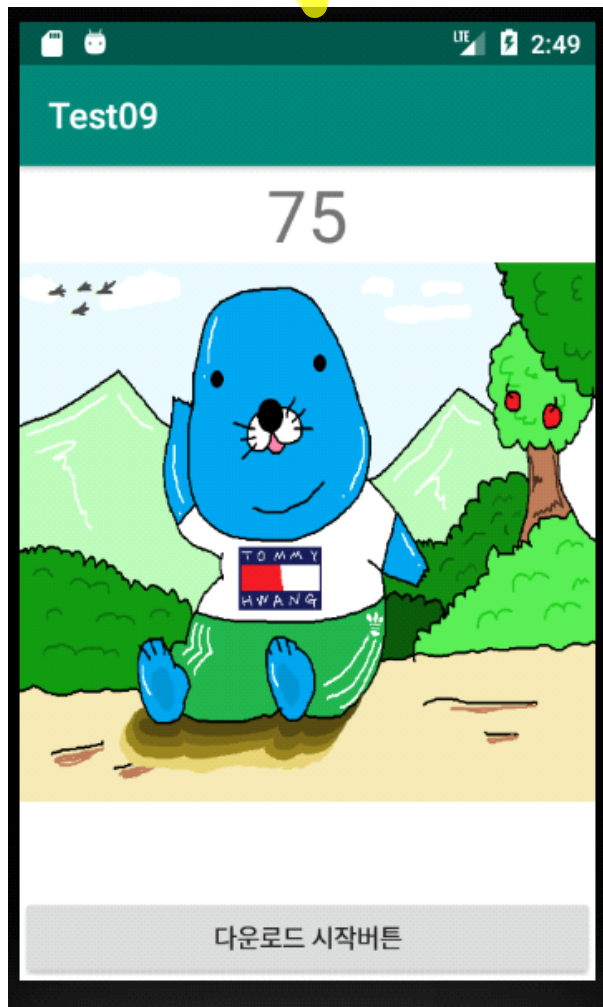
MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    private TextView tvCount;
    private ImageView imageDown;
    private Button btnDown;

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

        tvCount = findViewById(R.id.tv_count);
        imageDown = findViewById(R.id.image_download);
        btnDown = findViewById(R.id.btn_download);

        btnDown.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //다운로드 시작!
            }
        });
    }
}
```

12 Test10 api

2018년 11월 9일 금요일 오전 9:21

NAVER Developers

ProductsDocumentsApplicationNAVER D2Support

API 상태

Search Here

내 애플리케이션

애플리케이션 등록

Clova Platform Console

API 재휴 신청

계정 설정

bapsi

네이버 검색

애플리케이션의 기본 정보를 등록하면, 좌측 **내 애플리케이션** 메뉴의 서브 메뉴에 등록하신 애플리케이션 이름으로 서브 메뉴가 만들어집니다.

| | |
|--------------------|--|
| 애플리케이션 이름 | <div>아무거나 구별할수있는 이름</div> <div><ul style="list-style-type: none">네이버 아이디로 로그인할 때 사용자에게 표시되는 이름이므로 가급적 10자 이내의 간결한 이름을 사용해주세요.40자 이내의 영문, 한글, 숫자, 공백문자, "-", "_" 만 입력 가능합니다.</div> |
| 사용 API | <div>선택하세요.</div> <div>검색</div> |
| 비로그인 오픈 API 서비스 환경 | <div>환경 추가</div> <div>Android 설정</div> <div>안드로이드 앱 패키지 이름</div> <div>패키지명</div> <div>안드로이드 앱 패키지 이름을 입력하세요. 예)) com.example.mynavermap</div> |

- [애플리케이션 이름] 설정을 확인해 주세요.
- [안드로이드 앱 패키지 이름] 설정을 확인해 주세요.

Documents > 서비스 API > 검색 > 블로그

API 공통 가이드

SDK & Tools

Clova

네이버 아이디로 로그인

지도

파파고

서비스 API

데이터랩

검색

- 블로그

- 뉴스

- 책

- 성인 검색어 판별

검색 > 블로그

네이버 블로그 검색 결과를 출력해주는 REST API입니다. 비로그인 오픈 API이므로 GET으로 호출할 때 HTTP Header에 애플리케이션 시 발급받은 **Client ID**와 **Client Secret** 값을 같이 전송해 주시면 활용 가능합니다.

오픈 API 이용 신청 >

0.API 호출 예제

예제 실행 전에 아래 **1.준비사항** 항목들을 꼭 체크하시길 바랍니다.

| Java | PHP | Node.js | Python | C# |
|------|-----|---------|--------|----|
|------|-----|---------|--------|----|

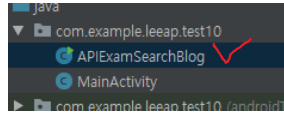
네이버 검색 API예제는 블로그를 비롯 전문자료까지 호출방법이 동일하므로 blog검색만 대표로 예제를 올렸습니다.

- 뉴스
- 책
- 성인 검색어판별
- 백과사전
- 영화
- 카페글
- 지식IN

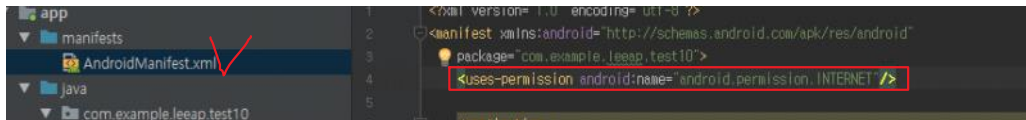
네이버 검색 API예제는 블로그를 비롯 전문자료까지 호출방법이 동일하므로 blog검색만 대표로 예제를 올렸습니다.
 // 네이버 검색 API 예제 - blog 검색

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
```

전체 복사 한 후 클래스 자체를 붙여넣기



```
public static void main(String[] args) {
    String clientId = "3v1kj90F2XIVTsbZyPTu"; //애플리케이션 클라이언트 아이디값;
    String clientSecret = "oTNSHH6er8"; //애플리케이션 클라이언트 시크릿값;
    try {
        String text = URLEncoder.encode("안드로이드", "UTF-8");
        String apiURL = "https://openapi.naver.com/v1/search/book?query="+ text; // json 결과
        //String apiURL = "https://openapi.naver.com/v1/search/blog.xml?query="+ text; // xml 결과
        URL url = new URL(apiURL);
        HttpURLConnection con = (HttpURLConnection)url.openConnection();
        con.setRequestMethod("GET");
        con.setRequestProperty("X-Naver-Client-Id", clientId);
        con.setRequestProperty("X-Naver-Client-Secret", clientSecret);
        int responseCode = con.getResponseCode();
        BufferedReader br;
```



error: android.os.NetworkOnMainThreadException

```
public class APIExamSearchBlog extends Thread {
    @Override
    public void run() {
        String clientId = "3v1kj90F2XIVTsbZyPTu"; //애플리케이션 클라이언트 아이디값;
        String clientSecret = "oTNSHH6er8"; //애플리케이션 클라이언트 시크릿값;
        try {
            String text = URLEncoder.encode("안드로이드", "UTF-8");
            String apiURL = "https://openapi.naver.com/v1/search/book?query="+ text; // json 결과
            //String apiURL = "https://openapi.naver.com/v1/search/blog.xml?query="+ text; // xml 결과
            URL url = new URL(apiURL);
            HttpURLConnection con = (HttpURLConnection)url.openConnection();
            con.setRequestMethod("GET");
            con.setRequestProperty("X-Naver-Client-Id", clientId);
            con.setRequestProperty("X-Naver-Client-Secret", clientSecret);
            int responseCode = con.getResponseCode();
            BufferedReader br;
            if(responseCode==200) { // 정상 호출
                br = new BufferedReader(new InputStreamReader(con.getInputStream()));
            } else { // 에러 발생
                br = new BufferedReader(new InputStreamReader(con.getErrorStream()));
            }
            String inputLine;
            StringBuffer response = new StringBuffer();
            while ((inputLine = br.readLine()) != null) {
                response.append(inputLine);
            }
        }
    }
}
```



```
activity_main.xml x MainActivity.java x AndroidManifest.xml x
package com.example.leeap.test10;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

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

        new APIExamSearchBlog().start();
    }
}
```

```
ample.leeap.test10 I/Ije error: android.os.NetworkOnMainThreadException
ample.leeap.test10 I/Ije: {"lastBuildDate": "Fri, 09 Nov 2018 09:52:12 +0900", "total": 1806, "start": 1, "display": 10, "items": [{"title": "Do it! <b>안드로이드</b> 앱 프로그래밍 (<b>안드로이드</b>"}
ample.leeap.test10 I/Ije: {"lastBuildDate": "Fri, 09 Nov 2018 09:52:35 +0900", "total": 1806, "start": 1, "display": 10, "items": [{"title": "Do it! <b>안드로이드</b> 앱 프로그래밍 (<b>안드로이드</b>"}

```

parsing

2018년 11월 9일 금요일 오전 10:08

1. Xml 레이아웃 파일 작성

- : item_book.xml(ImageView 1, TextView 4)
- : Activity_main.xml : EditText, Button, ListView

2. Java 파일 작성

- : BookVO클래스 작성
- : NaverThread 클래스 작성 (메인 스레드는 네트워크 하면 안되니까)
- : MainActivity 검색버튼 이벤트
- : BookAdapter 클래스 작성(스레드가 데이터 받은 이후에 본격적으로 리스트뷰에 작업하는 애)

parsing

```
layout
  activity_main.xml
  item_book.xml ✓

<ImageView
    android:layout_width="40dp"
    android:layout_height="60dp"
    android:id="@+id/book_img"
    android:layout_marginLeft="10dp"
    app:layout_constraintLeft_toLeftOf="parent"
    android:background="#550000ff"/>

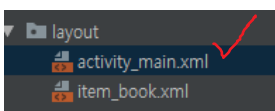
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="책 제목 샘플"
    android:layout_marginLeft="10dp"
    app:layout_constraintLeft_toRightOf="@+id/book_img"
    android:textSize="15sp"
    android:textStyle="bold"
    android:id="@+id/book_tv_title"/>

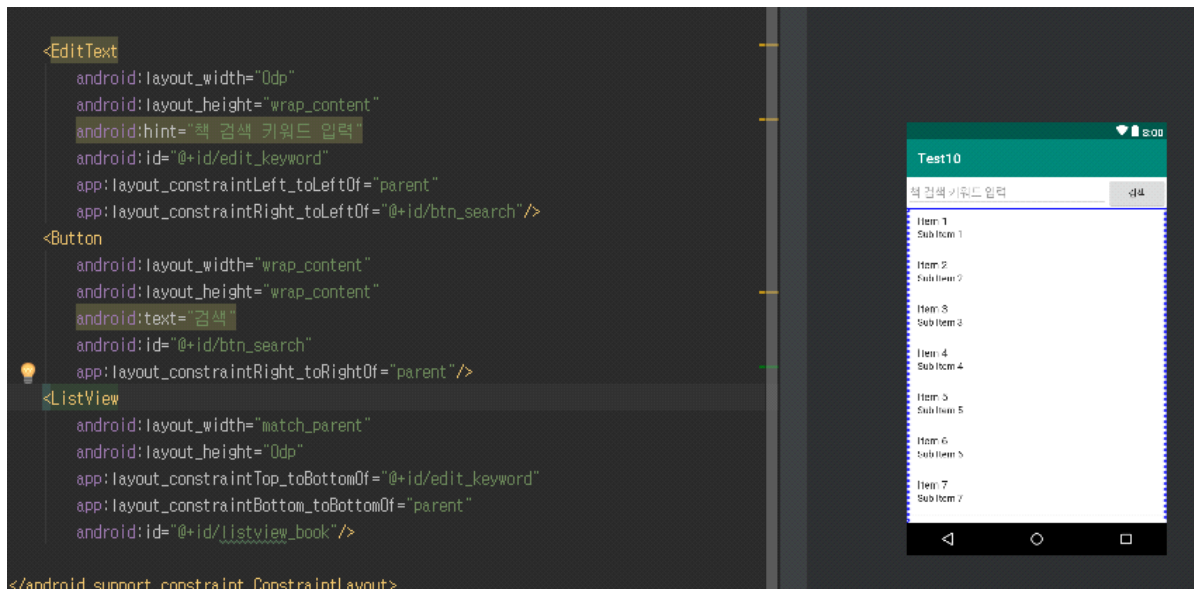
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/book_tv_pub"
    android:text="출판사 샘플"
    android:layout_marginLeft="10dp"
    app:layout_constraintLeft_toRightOf="@+id/book_img"
    app:layout_constraintTop_toBottomOf="@+id/book_tv_title"/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/book_tv_writer"
    android:text="저자 이름"
    android:layout_marginLeft="10dp"
    app:layout_constraintLeft_toRightOf="@+id/book_img"
    app:layout_constraintTop_toBottomOf="@+id/book_tv_pub"/>

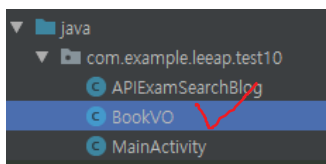
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/book_tv_price"
    android:text="10000"
    app:layout_constraintTop_toBottomOf="@+id/book_tv_writer"
    app:layout_constraintRight_toRightOf="parent"
    android:layout_marginRight="20dp"/>

android.support.constraint.ConstraintLayout>
android.support.constraint.ConstraintLayout>
```





VO class



Getter setter 생성

```
public class BookVO {
    private String imgUrl; //책 사진 얻을 수 있는 url 주소
    private String title;
    private String publisher;
    private String writer;
    private String price;
    /////////////// alr+insert ///////////////////////////////////
    public String getImgUrl() {
        return imgUrl;
    }

    public void setImgUrl(String imgUrl) {
        this.imgUrl = imgUrl;
    }

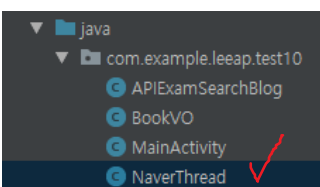
    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getPublisher() {
        return publisher;
    }

    public void setPublisher(String publisher) {
        this.publisher = publisher;
    }
}
```

NaverThread class



```

public class NaverThread extends Thread{
    private String keyword;

    public NaverThread(String keyword){
        this.keyword = keyword;
    }

    @Override
    public void run() {
        String clientId = "3v1k190F2XIVTsbZyPTu";//애플리케이션 클라이언트 아이디값";
        String clientSecret = "oTNSHH6er8";//애플리케이션 클라이언트 시크릿값";
        try {
            String text = URLEncoder.encode(keyword, "UTF-8");
            String apiURL = "https://openapi.naver.com/v1/search/book?query=" + text; // json 결과
            //String apiURL = "https://openapi.naver.com/v1/search/blog.xml?query=" + text; // xml 결과
            URL url = new URL(apiURL);
            HttpURLConnection con = (HttpURLConnection) url.openConnection();
            con.setRequestMethod("GET");
            con.setRequestProperty("X-Naver-Client-Id", clientId);
            con.setRequestProperty("X-Naver-Client-Secret", clientSecret);
            int responseCode = con.getResponseCode();
            BufferedReader br;
            if (responseCode == 200) { // 정상 호출
                br = new BufferedReader(new InputStreamReader(con.getInputStream()));
            } else { // 에러 발생
                br = new BufferedReader(new InputStreamReader(con.getErrorStream()));
            }
            String inputLine;
            StringBuffer response = new StringBuffer();
            while ((inputLine = br.readLine()) != null) {
                response.append(inputLine);
            }
            br.close();
            //System.out.println(response.toString());
            Log.i("tag", "Ije", response.toString());
        } catch (Exception e) {
            //System.out.println(e);
            Log.i("tag", "Ije error", e.toString());
        }
    }
}

```

Gson repository

← → ↻ 🔒 https://mvnrepository.com/search?q=gson



| Repository | Found 281 results |
|---|--|
| <ul style="list-style-type: none"> Central 185 Sonatype 120 JCenter 61 Spring Plugins 54 Spring Lib M 51 IBiblio 10 Spring Lib Release 9 | <p>Sort: relevance popular newest</p> <div>  <div> <p>1. Gson</p> <p>com.google.code.gson » gson</p> <p>Gson</p> <p>Last Release on May 22, 2018</p> </div> <div> <p>8,678 usages</p> <p>Apache</p> </div> </div> |



Gson » 2.8.0

Gson

| | |
|--------------|--|
| License | Apache 2.0 |
| Categories | JSON Libraries |
| Date | (Oct 27, 2016) |
| Files | pom (1 KB) jar (226 KB) View All |
| Repositories | Central Mulesoft Sonatype |
| Used By | 8,678 artifacts |

Note: There is a new version for this artifact

New Version

2.8.5

Maven

Gradle

SBT

Ivy

Grape

Leiningen

Buildr

```
<!-- https://mvnrepository.com/artifact/com.google.code.gson/gson -->
<dependency>
  <groupId>com.google.code.gson</groupId>
  <artifactId>gson</artifactId>
  <version>2.8.0</version>
</dependency>
```

☒ Include comment with link to declaration

Maven

Gradle

SBT

Ivy

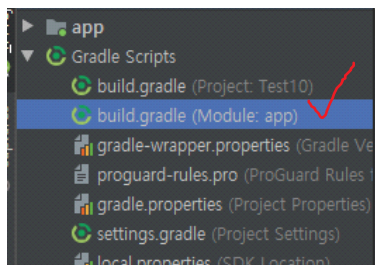
Grape

Leiningen

Buildr

복사

```
// https://mvnrepository.com/artifact/com.google.code.gson/gson
compile group: 'com.google.code.gson', name: 'gson', version: '2.8.0'
```



```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
```

```
// https://mvnrepository.com/artifact/com.google.code.gson/gson
compile group: 'com.google.code.gson', name: 'gson', version: '2.8.0'
```

```
// https://mvnrepository.com/artifact/com.google.code.gson/gson
implementation group: 'com.google.code.gson', name: 'gson', version: '2.8.0'
```

NaverThread.java

```
@Override
public void run() {
```

```
while ((inputLine = br.readLine()) != null) {
    response.append(inputLine);
}
br.close();
//System.out.println(response.toString());
```

```
@Override
public void run() {
    String naverResult = null;

    String clientId = "2x1k190E2Y1W"
```

```
public class NaverThread extends Thread{
    private String keyword;
    private Handler mainHandler;

    public NaverThread(String keyword, Handler handler){
        this.keyword = keyword;
        this.mainHandler = handler;
    }
}
```

```
        response.append(inputLine);
    }
    br.close();
    //System.out.println(response.toString());
    naverResult = response.toString(); //네이버가 응답한 검색 결과 데이터

    // gson 라이브러리를 이용해서 json->java object로 파싱하기
    Gson gson = new Gson();
    BookVO[] array = gson.fromJson(naverResult, BookVO[].class);
    ArrayList<BookVO> bookList = (ArrayList<BookVO>)Arrays.asList(array);

} catch (Exception e) {
    //System.out.println(e);
    Log.i( tag: "Ije error", e.toString());
}
}
```

```
//결과 리스트를 MainActivity한테 보내야 함.
Message msg = new Message();
msg.obj = bookList;
mainHandler.sendMessage(msg);
```

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    private EditText editKeyword;
    private Button btnSearch;
    private ListView listViewBook;

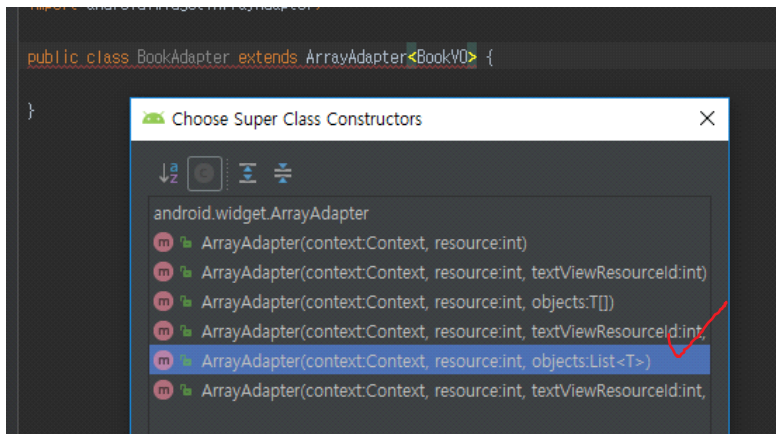
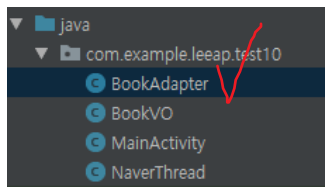
    private Handler handler = new Handler(){
        @Override
        public void handleMessage(Message msg) {
            //NaverThread가 작업을 마치면 여기서 메시지를 보냄
        }
    }
}
```

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

    editKeyword = findViewById(R.id.edit_keyword);
    btnSearch = findViewById(R.id.btn_search);
    listViewBook = findViewById(R.id.listview_book);

    btnSearch.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String keyword = editKeyword.getText().toString();
            if(keyword==null || keyword.length()==0){
                Toast.makeText( context MainActivity.this, text: "검색어를 입력하세요", Toast.LENGTH_SHORT).show();
            }else {
                new NaverThread(keyword, handler).start(); //스레드가 네이버한테 검색어 보내서 결과 받아 올거야
            }
        }
    });
}
```

BookAdapter.java



```
public class BookAdapter extends ArrayAdapter<BookVO> {  
    }  
  
public class BookAdapter extends ArrayAdapter<BookVO> {  
    private Activity context;  
    private int itemLayout; //R.layout.item_book;  
    private ArrayList<BookVO> bookList;  
  
    public BookAdapter(Context context, int resource, List<BookVO> objects) {  
        super(context, resource, objects);  
  
        this.context = (Activity) context;  
        this.itemLayout = resource;  
        this.bookList = (ArrayList<BookVO>) objects;  
    }  
  
    @Override  
    public View getView(int position, View convertView, ViewGroup parent) {  
        BookHolder holder = null;  
  
        if (convertView == null) { //getView 첫 실행 -> inflate  
            LayoutInflater inflater = context.getLayoutInflater();  
            convertView = inflater.inflate(itemLayout, parent, attachToRoot false);  
  
            holder = new BookHolder();  
            holder.bookImageView = convertView.findViewById(R.id.book_img);  
            holder.bookTitle = convertView.findViewById(R.id.book_tv_title);  
            holder.bookPub = convertView.findViewById(R.id.book_tv_pub);  
            holder.bookWriter = convertView.findViewById(R.id.book_tv_writer);  
            holder.bookPrice = convertView.findViewById(R.id.book_tv_price);  
  
            convertView.setTag(holder);  
        } else {  
            //이전에 inflate한 적 있네? 그거 갖다 써야지  
            holder = (BookHolder) convertView.getTag();  
        }  
  
        BookVO bookVO = bookList.get(position); //책 한권 뽑기  
  
        holder.bookTitle.setText(bookVO.getTitle());  
        holder.bookPub.setText(bookVO.getPublisher());  
        holder.bookWriter.setText(bookVO.getWriter());  
        holder.bookPrice.setText(bookVO.getPrice());  
  
        return super.getView(position, convertView, parent);  
    }  
  
    class BookHolder {  
        ImageView bookImageView;  
        TextView bookTitle;  
        TextView bookPub;  
        TextView bookWriter;  
        TextView bookPrice;  
    }  
}
```

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    private EditText editKeyword;
    private Button btnSearch;
    private ListView listViewBook;
    private ArrayList<BookVO> bookList = new ArrayList<>(); //비어있는책 데이터 리스트(스레드가 받아올 예정)
    private BookAdapter adapter;

    private Handler handler = new Handler(){
        @Override
        public void handleMessage(Message msg) {
            //NaverThread가 작업을 마치면 여기로 메시지를 보냄!
            bookList.clear();
            bookList.addAll((ArrayList<BookVO>)msg.obj);

            adapter.notifyDataSetChanged(); //어댑터한테 데이터 온거 알려주기
        }
    };

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

        editKeyword = findViewById(R.id.edit_keyword);
        btnSearch = findViewById(R.id.btn_search);
        listViewBook = findViewById(R.id.listView_book);

        btnSearch.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String keyword = editKeyword.getText().toString();
                if(keyword==null||keyword.length()==0){
                    Toast.makeText(context, MainActivity.this, "검색어를 입력하세요", Toast.LENGTH_SHORT).show();
                } else {
                    new NaverThread(keyword, handler).start(); //스레드가 네이버한테 검색어 보내서 결과 받아 올거야
                }
            }
        });

        adapter = new BookAdapter(context, this, R.layout.item_book, bookList);
        listViewBook.setAdapter(adapter);
    }
}
```