

모바일프로그래밍

- 안드로이드(Android) -

- 안드로이드 기능/실습 - 19
 - Content Provider - Audio(미디어)
- 안드로이드 기능/실습 - 20
 - Content Provider - 통화

□ 안드로이드 기능/실습 - 19

CONTENT PROVIDER - AUDIO

- 애플리케이션 내에 저장되어 있는 데이터베이스
 - 해당 애플리케이션 외에 다른 애플리케이션이 접근하는 것이 불가능
 - 그러나, 데이터 공유 필요
- 데이터 공유
 - 콘텐츠 프로바이더 제공
- 콘텐츠 프로바이더
 - 애플리케이션이 보유하고 있는 데이터를 다른 애플리케이션에서 공유하도록 공개하기 위한 모듈
 - 모든 패키지들이 데이터를 공유하기 위해서는 콘텐츠 프로바이더를 사용해야 함
 - 기본적으로 제공되는 콘텐츠 프로바이더와 원하는 콘텐츠 프로바이더를 직접 개발해서 이용하는 커스텀 콘텐츠 프로바이더가 존재

□ Content Provider

컨텐츠 프로바이더	설명
CallLog	통화 기록
Browser	브라우저의 북마크, 히스토리
Contacts	연락처
MediaStore	외부 매체에 저장된 오디오, 비디오, ...
UserDictionary	사용자 정의 단어 사전

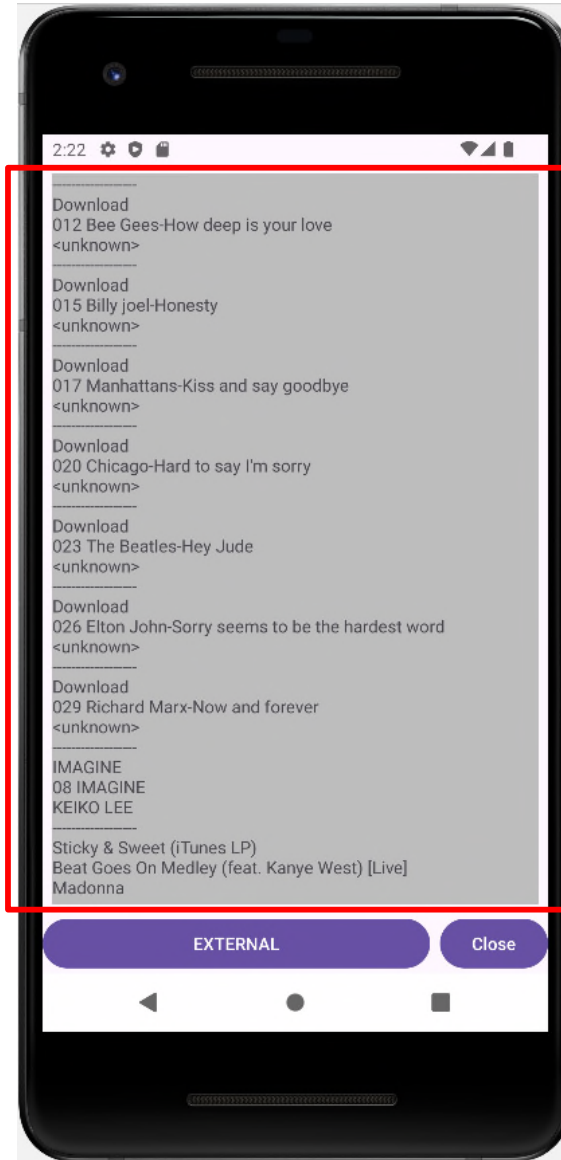
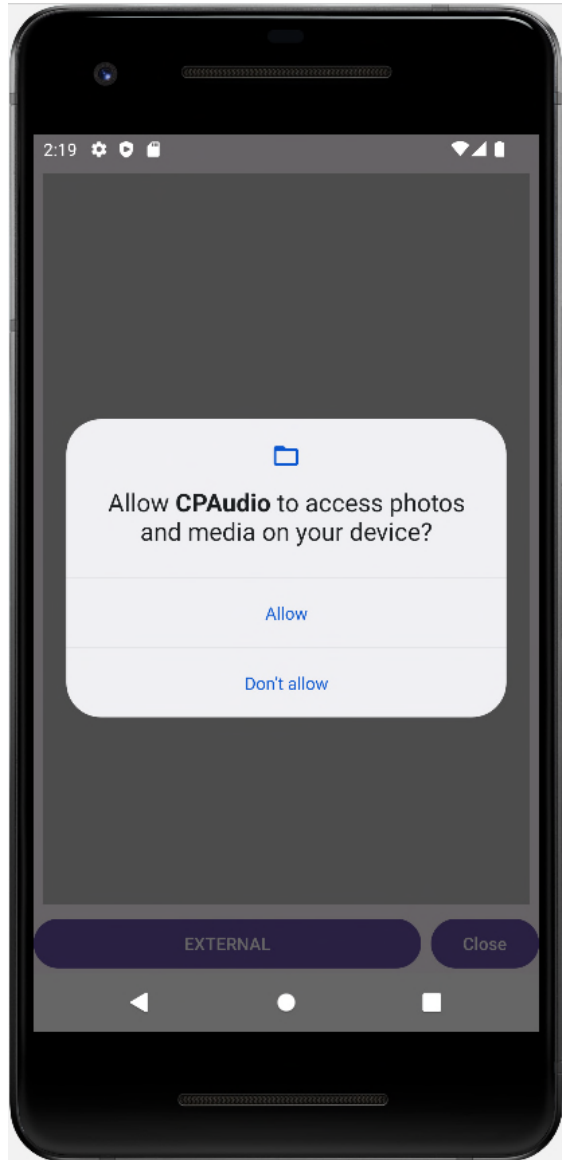
- 안드로이드 버전에 따라 기능이 제한된 경우가 있음
- sdk 타겟 버전을 6.0 이상으로 하는 경우, 권한 부여에 대한 처리가 추가로 필요

○ 콘텐츠 프로바이더를 사용하기 위해서는 해당 콘텐츠 프로바이더의 주소를 알아야 하며 콘텐츠 프로바이더에 접근할 때, 콘텐츠 리졸버가 필요함

URI	데이터
CallLog.Calls.CONTENT_URI	통화 기록
ContactsContract.Contacts.CONTENT_URI ...	연락처 목록 ...
MediaStore.Image.Media.INTERNAL_CONTENT_URI	내장 미디어 이미지
MediaStore.Image.Media.EXTERNAL_CONTENT_URI	외장 미디어 이미지
MediaStore.Video.Media.INTERNAL_CONTENT_URI	내장 미디어 비디오
MediaStore.Video.Media.EXTERNAL_CONTENT_URI	외장 미디어 비디오
MediaStore.Audio.Media.INTERNAL_CONTENT_URI	내장 미디어 오디오
MediaStore.Audio.Media.EXTERNAL_CONTENT_URI	외장 미디어 오디오

- Mediastore.Audio.Media 클래스
 - 오디오 파일을 관리하는 클래스
 - 오디오 파일의 제목이나 재생 길이 등을 얻는 것이 가능
- 내장
 - MediaStore.Audio.Media.INTERNAL_CONTENT_URI
- 외장
 - MediaStore.Audio.Media.EXTERNAL_CONTENT_URI
- 테스트 전에 메모리에 데이터 추가 과정 진행

○외장 미디어의 오디오(음악파일) 정보를 추출하여 출력

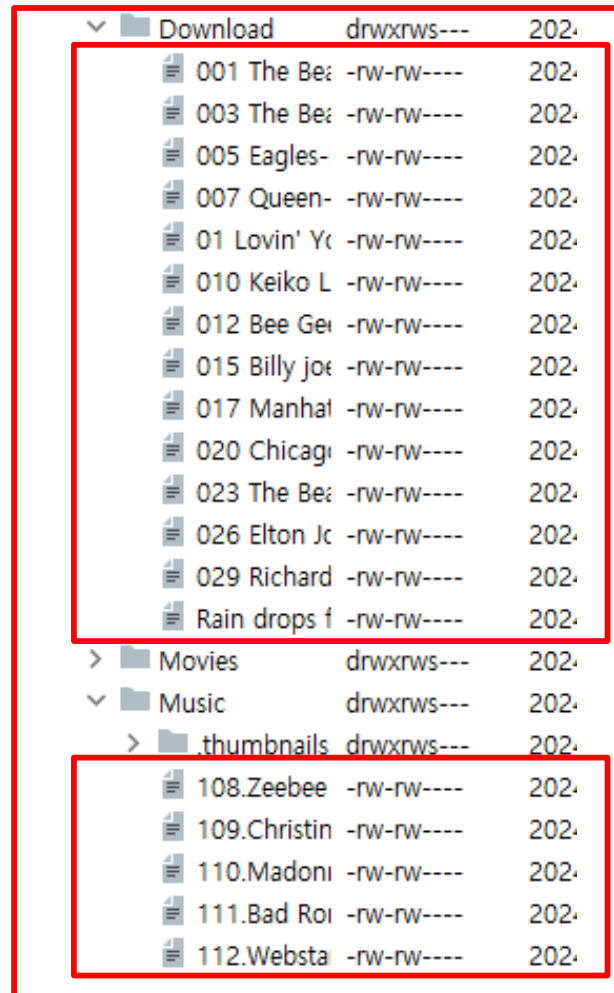


음악파일의 정보를 출력
-타이틀
-앨범
-아티스트 ...

□ 외장메모리에 파일 추가하는 과정

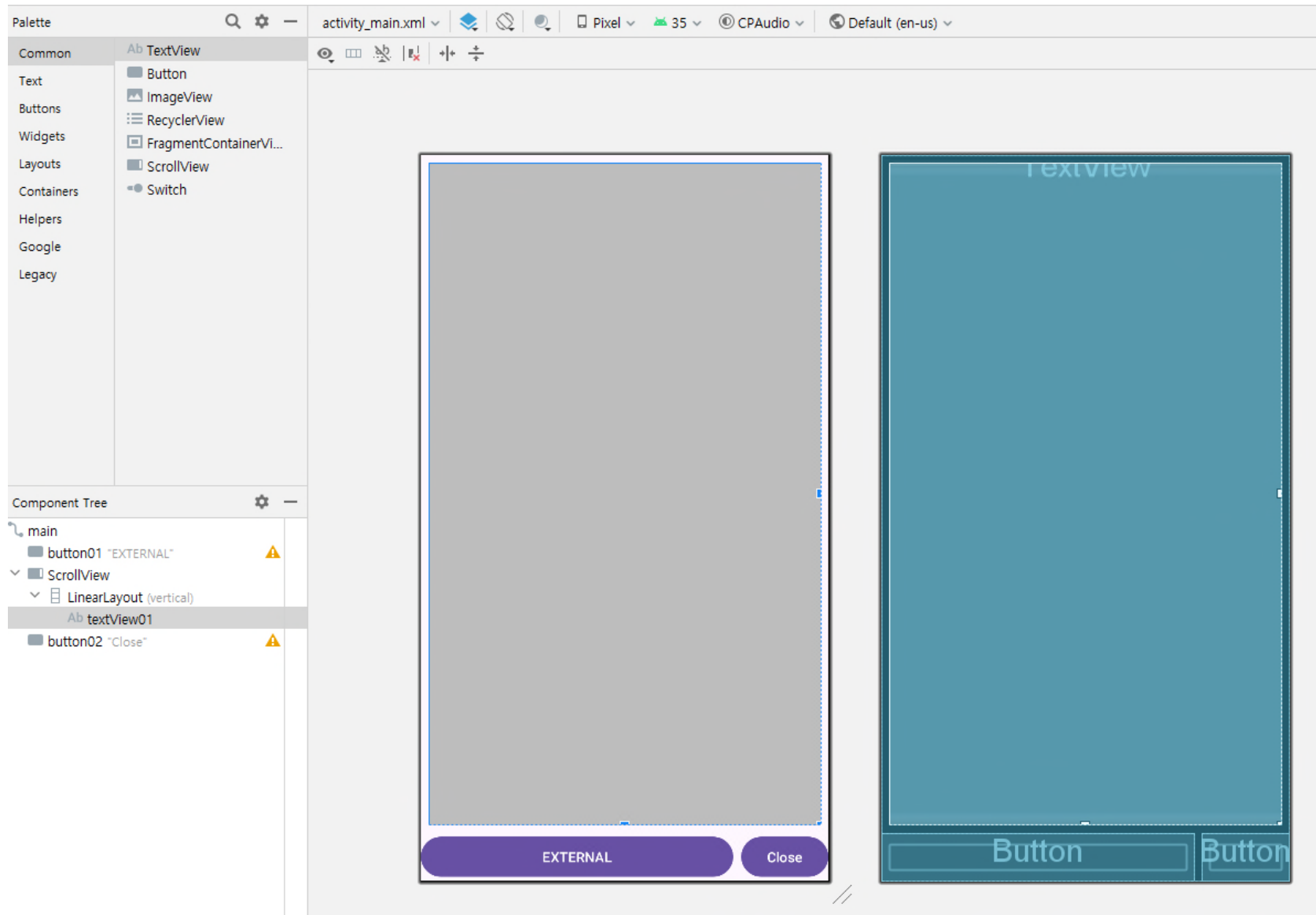
○ Device File Explorer에서 해당 위치에 파일 추가

- storage/ 하단
- 위치는 버전에 따라 차이



원하는 위치에 데이터 추가
추가는 우클릭, 업로드 선택

□ 레이아웃 구성



□ 레이아웃 코드

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/button01"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:text="EXTERNAL"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toStartOf="@+id/button02"
        app:layout_constraintStart_toStartOf="parent" />

    <Button
        android:id="@+id/button02"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Close"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />
```

□ 레이아웃 코드

오디오(음악파일) 정보를 출력할
TextView에 스크롤 기능을 추가

```
<ScrollView
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:fillViewport="true"
    app:layout_constraintBottom_toTopOf="@+id/button01"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical" >

        <TextView
            android:id="@+id/textView01"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:background="#FFBDBDBD" />

    </LinearLayout>
</ScrollView>

</androidx.constraintlayout.widget.ConstraintLayout>
```

오디오 정보를 출력할 TextView

□ AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"
        android:maxSdkVersion="32" />

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.CPAudio"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

외부 저장소 사용 권한 추가

- 안드로이드 버전에 따른 권한 변경이 있음 (추후 설명)
- API32 이상은 어떻게 처리해야 하는가에 대한 확인 (별도 실습으로 진행)
- 6.0 이상이 타겟 버전인 경우, 권한 추가 뿐만 아니라 권한에 따라서 (위험 권한의 경우) 권한허용을 위한 처리 코드가 별도로 필요

□ 코드 분석

```
package com.practice.ex.cpaudio;

import android.Manifest;
import android.content.ContentResolver;
import android.database.Cursor;
import android.os.Bundle;
import android.provider.MediaStore;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {
```

```
    TextView text01; 2 usages
```

```
    Button btn01, btn02; 2 usages
```

□ 코드 분석

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeToEdge.enable( $this$enableEdgeToEdge: this);
    setContentView(R.layout.activity_main);
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {...});

    btn01 = (Button) findViewById(R.id.button01);
    btn01.setOnClickListener(this);

    btn02 = (Button) findViewById(R.id.button02);
    btn02.setOnClickListener(this);

    text01 = (TextView) findViewById(R.id.textView01);

    ActivityCompat.requestPermissions( activity: this, new String[]
        {Manifest.permission.WRITE_EXTERNAL_STORAGE}, MODE_PRIVATE);
}
```

사용자에게 권한 요청 부분

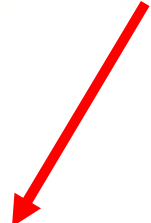
- 권한 확인을 통해 필요 시 동작하도록 구현
- 버전에 따라 사용 권한이 다른 경우, 버전 확인 및 별도 처리가 필요 (별도 실습 진행)
- 여기서는 기본 사용법을 보이기 위해 기본 코드만 작성

□ 코드 분석

```
@Override
public void onClick(View v) {
    if(v.getId() == R.id.button01) {
        ContentResolver cr = getContentResolver();
        Cursor c = cr.query(MediaStore.Audio.Media.EXTERNAL_CONTENT_URI, projection: null,
            selection: null, selectionArgs: null, sortOrder: MediaStore.Audio.Media.TITLE + " ASC");

        int albumIdx = c.getColumnIndexOrThrow(MediaStore.Audio.Media.ALBUM);
        int titleIdx = c.getColumnIndexOrThrow(MediaStore.Audio.Media.TITLE);
        int artistIdx = c.getColumnIndexOrThrow(MediaStore.Audio.Media.ARTIST);
    }
}
```

특정 필드의 인덱스값 반환



컨텐츠 프로바이더 URI,
조회할 컬럼,
선택할 줄,
선택 조건,
정렬 조건

□ 코드 분석

```
StringBuffer buff = new StringBuffer();

buff.append("Music List\n\n");

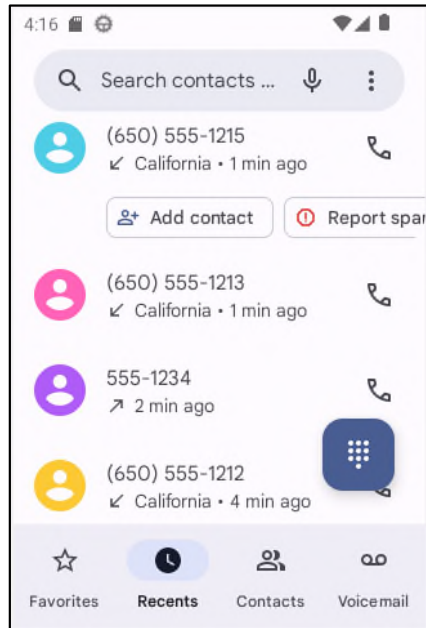
    if(c.moveToFirst()) {
        do {
            buff.append(c.getString(albumIdx));
            buff.append("\n");
            buff.append(c.getString(titleIdx));
            buff.append("\n");
            buff.append(c.getString(artistIdx));
            buff.append("\n");
            buff.append("-----\n");
        } while (c.moveToNext());
    }

    text01.setText(buff);
}
if(v.getId() == R.id.button02) {
    finish();
}
}
}
```

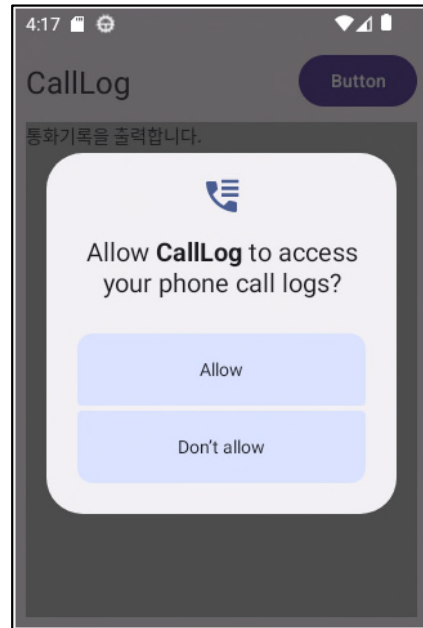
반환받은 인덱스값을 이용하여
원하는 순서, 모양 등으로
데이터 출력

□ 안드로이드 기능/실습 - 20
CONTENT PROVIDER - 통화

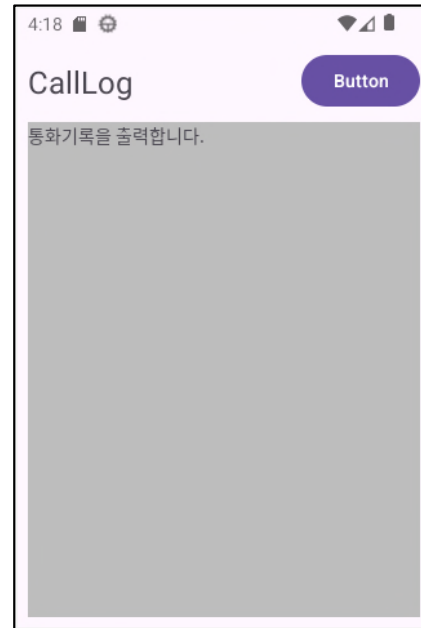
□ 실행 모습



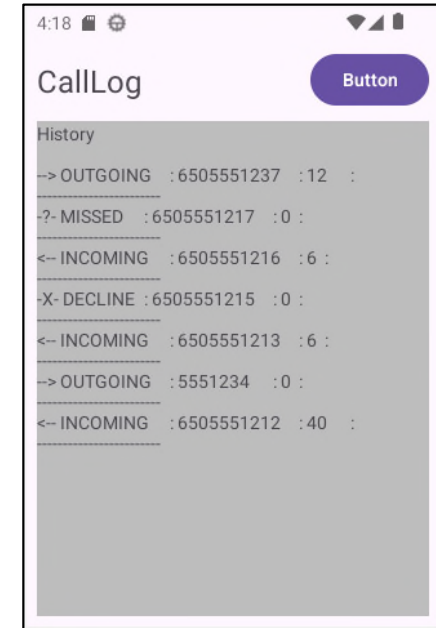
통화내역 생성/확인



권한 확인



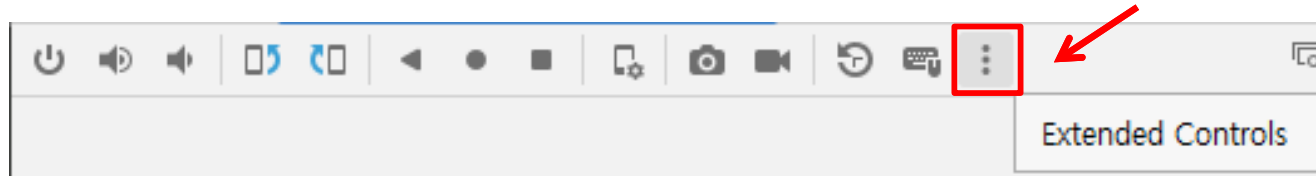
초기실행 모습



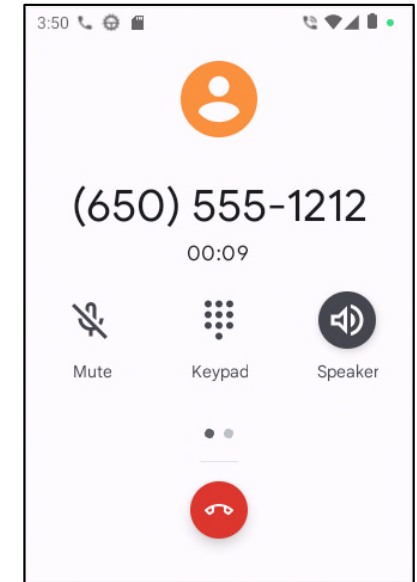
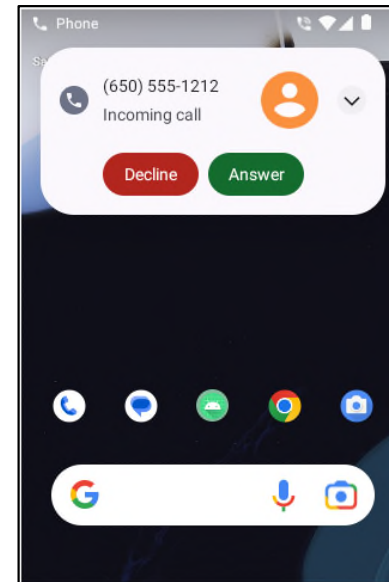
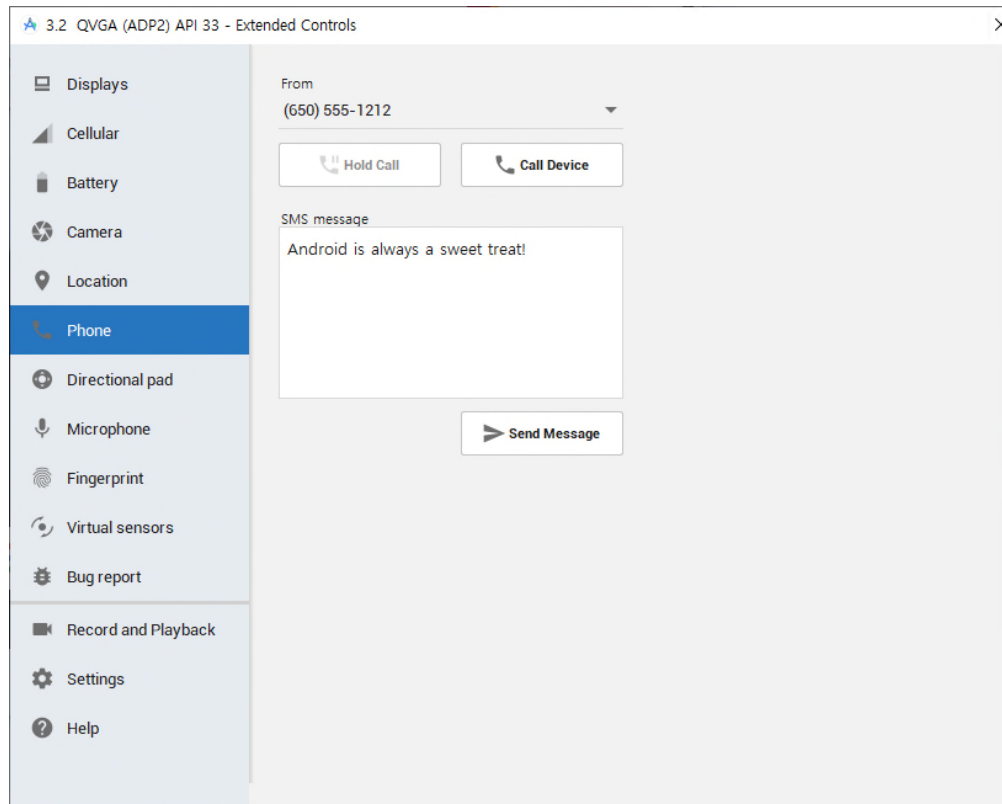
통화기록 출력

□ 통화 내역 생성

○에뮬레이터에서 통화 내역을 테스트하기 위해 통화내역을 생성

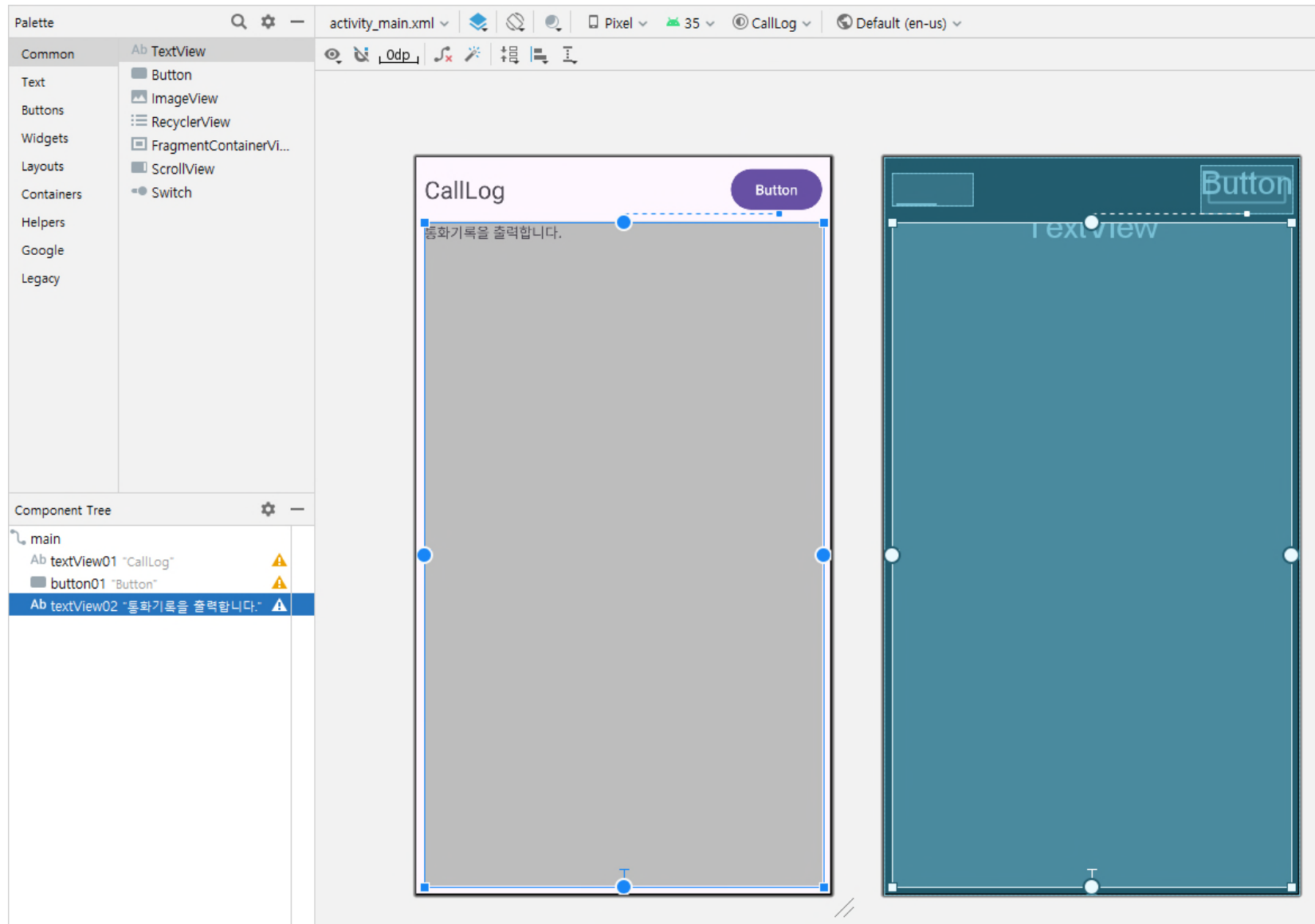


○Extended Controls을 이용하여 통화내역 생성



□ 레이아웃 구성

○ 통화내역을 호출할 버튼과 결과를 출력할 텍스트뷰로 구성



□ 레이아웃 코드

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
```

```
<TextView
    android:id="@+id/textView01"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginBottom="8dp"
    android:text="CallLog"
    android:textSize="24sp"
    app:layout_constraintBottom_toTopOf="@+id/textView02"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

상단에 표시되는
TextView

□ 레이아웃 코드

```
<Button
    android:id="@+id/button01"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:text="Button"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
    android:id="@+id/textView02"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:background="#BDBDBD"
    android:text="통화기록을 출력합니다."
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button01" />
```

CallLog를 기록할 텍스트뷰

* 스크롤이 필요한 경우,
스크롤뷰 추가하여 구성

```
</androidx.constraintlayout.widget.ConstraintLayout>
```


□ *AndroidManifest.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <uses-permission android:name="android.permission.READ_CALL_LOG"/>

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="CallLog"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.CallLog"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

권한 추가

- 6.0 이상이 타겟 버전인 경우,
권한 추가 뿐만 아니라 권한에 따라서
(위험 권한의 경우) 권한허용을 위한
처리 코드가 별도로 필요

□ 코드 분석

```
package com.practice.ex.calllog;

import android.Manifest;
import android.content.ContentResolver;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.os.Bundle;
import android.provider.CallLog;
import android.text.method.ScrollingMovementMethod;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {
```

```
Cursor c; 10 usages
ContentResolver cr; 2 usages
String[] projection; 2 usages
StringBuffer buff; 13 usages
TextView text01; 3 usages
Button btn01; 2 usages
```

□ 코드 분석

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    EdgeToEdge.enable( $this$enableEdgeToEdge: this);  
    setContentView(R.layout.activity_main);  
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {...});
```

```
projection = new String[] {  
    CallLog.Calls.TYPE,  
    CallLog.Calls.NUMBER,  
    CallLog.Calls.DURATION  
};
```

조회하고자 할 컬럼 내용 작성

- 착, 발신 타입
- 전화번호
- 통화시간

```
text01 = (TextView) findViewById(R.id.textView02);  
text01.setMovementMethod(new ScrollingMovementMethod());
```

TextView에 스크롤 기능 추가

```
btn01 = (Button) findViewById(R.id.button01);  
btn01.setOnClickListener(this);
```

```
buff = new StringBuffer();  
buff.append("History \n\n");
```

사용자에게 권한 요청 부분
- READ_CALL_LOG 권한 요청

```
ActivityCompat.requestPermissions( activity: this,  
    new String[]{Manifest.permission.READ_CALL_LOG}, MODE_PRIVATE);
```

□ 코드 분석

@Override

```
public void onClick(View v) {
```

```
    cr = getContentViewResolver();
```

필요한 경우, 권한확인을 통해서
사용자에게 권한을 획득했는지 확인

```
    if(ActivityResultCompat.checkSelfPermission( context: this,
        Manifest.permission.READ_CALL_LOG)
        != PackageManager.PERMISSION_GRANTED) {

        //      example :
        //      ActivityCompat.requestPermissions(this, new String[]
        //          {Manifest.permission.READ_CALL_LOG}, MODE_PRIVATE);

        return;
    }
```

```
    c = cr.query(CallLog.Calls.CONTENT_URI, projection,
        selection: null, selectionArgs: null, sortOrder: "date desc");
```

컨텐츠 리졸버 인스턴스를 얻어와 query() 메소드를
호출한 결과를 커서로 얻어옴

{ 컨텐츠 프로바이더 URI,
조회할 컬럼,
선택할 줄,
선택 조건,
정렬 조건

□ 코드 분석

```
if(c.moveToFirst()) {  
    do {  
        if(c.getInt( columnIndex: 0) == CallLog.Calls.INCOMING_TYPE) {  
            buff.append("<-- INCOMING");  
        }  
        if(c.getInt( columnIndex: 0) == CallLog.Calls.MISSED_TYPE) {  
            buff.append("-?- MISSED");  
        }  
        if(c.getInt( columnIndex: 0) == CallLog.Calls.REJECTED_TYPE) {  
            buff.append("-X- DECLINE");  
        }  
        if(c.getInt( columnIndex: 0) == CallLog.Calls.OUTGOING_TYPE) {  
            buff.append("--> OUTGOING");  
        }  
  
        buff.append("\t : ");  
        buff.append(c.getString( columnIndex: 1));  
        buff.append("\t : ");  
        buff.append(c.getInt( columnIndex: 2));  
        buff.append("\t : \n");  
        buff.append("-----\n");  
    } while(c.moveToNext());  
}  
  
c.close();  
  
text01.setText(buff);  
}
```

착, 발신, 거절, 부재중 분류
- 실제 안드로이드폰에서 구현할 경우, 테스트 필요. 코드 등 정보 관리 부분이 차이가 있을 수 있음

통화번호, 통화시간 추출
데이터 형에 대응할 메소드를 사용