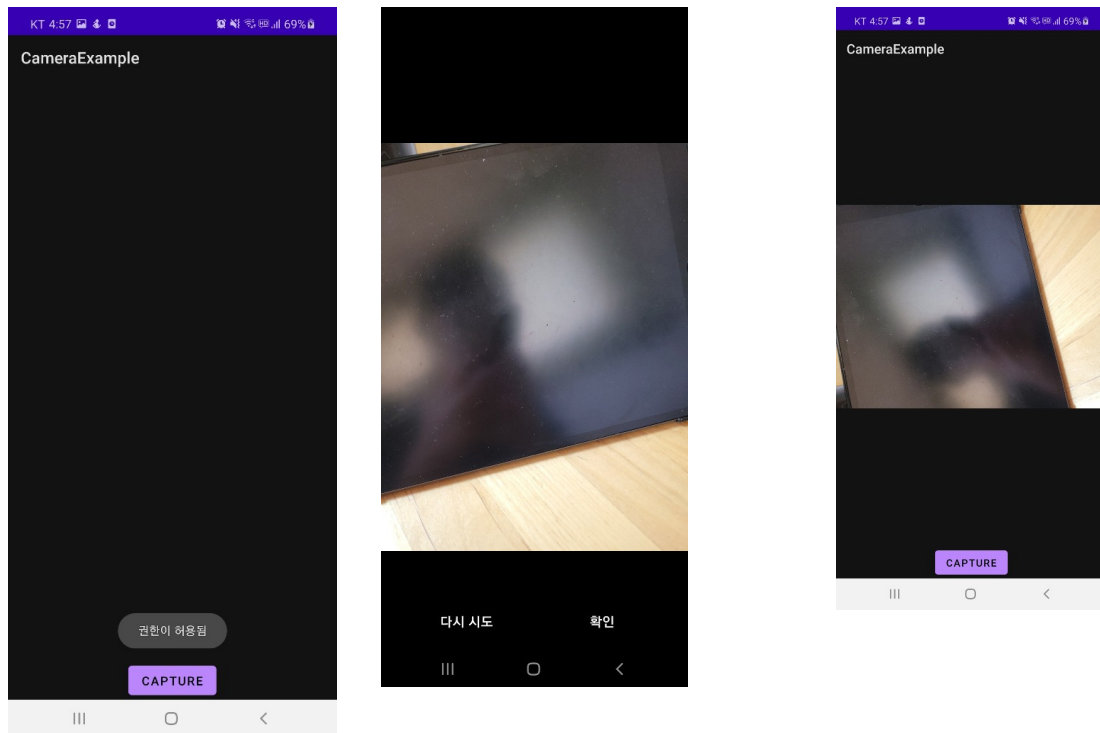


# 07 Camera 기능

## 결과



Capture 버튼을 누르면 카메라 권한 허용을 하라는 창이 뜨고 카메라가 작동한다. 사진 찍고 확인을 누르면 어플 화면 ImageView 에 사진이 뜬다.

## 1. activity\_main.xml 코드

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

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1">

        <ImageView
            android:id="@+id/iv_result"
            android:layout_width="match_parent"
            android:layout_height="match_parent"/>
    </LinearLayout>

    <LinearLayout
        android:orientation="horizontal"
        android:gravity="center"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <Button
```

```

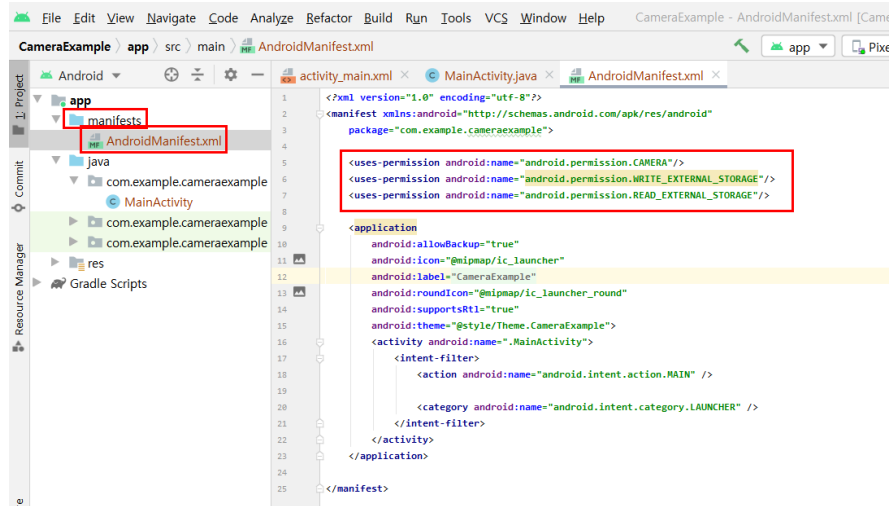
        android:id="@+id/btn_capture"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Capture" />

    </LinearLayout>

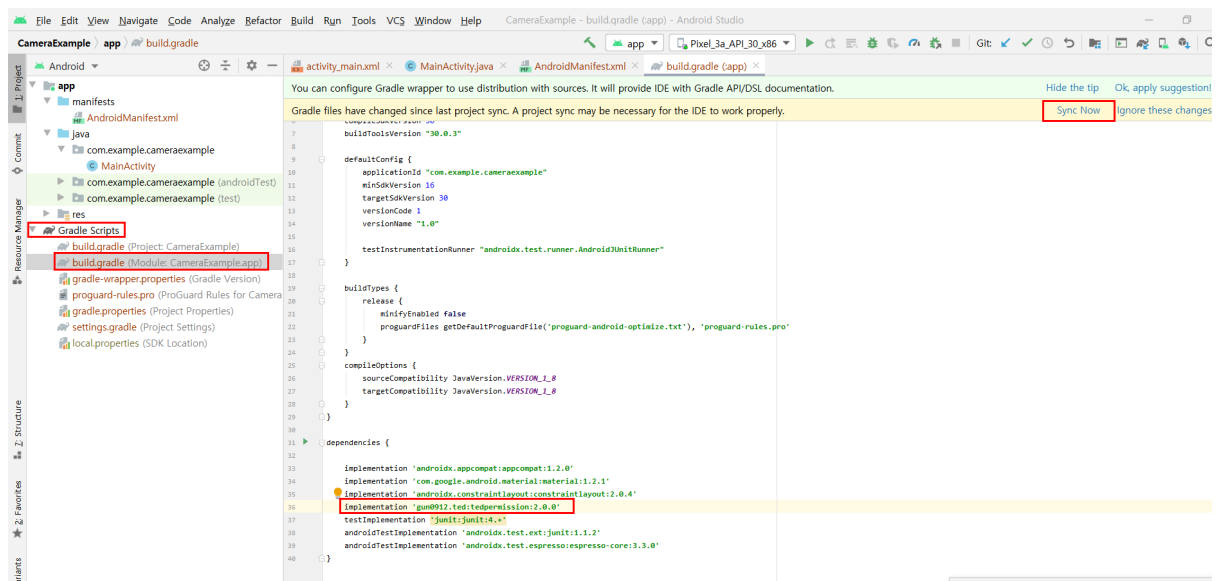
</LinearLayout>

```

## 2. AndroidManifest.xml에서 user-permission 추가



## 3. build.gradle (app)에서 'gun0912.ted:tedpermission:2.0.0' 추가 및 다운



다운로드하면 안드로이드 권한 허용을 할 때 권한 묻는 창 만들기 쉬움

## 4. MainActivity.java 코드

```
package com.example.cameraexample;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.content.FileProvider;

import android.Manifest;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Matrix;
import android.media.ExifInterface;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.provider.MediaStore;
import android.view.View;
import android.widget.ImageView;
import android.widget.Toast;

import com.gun0912.tedpermission.PermissionListener;
import com.gun0912.tedpermission.TedPermission;

import java.io.File;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;

public class MainActivity extends AppCompatActivity {

    private static final int REQUEST_IMAGE_CAPTURE = 672;
    private String imageFilePath;
    private Uri photoUri;

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

        // 권한 체크
        TedPermission.with(getApplicationContext())
            .setPermissionListener(permissionListener)
            .setRationaleMessage("카메라 권한이 필요합니다.")
            .setDeniedMessage("거부하셨습니다.")
            .setPermissions(Manifest.permission.WRITE_EXTERNAL_STORAGE, Manifest.permission.CAMERA)
            .check();

        findViewById(R.id.btn_capture).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
                if(intent.resolveActivity(getPackageManager()) != null) {
                    File photoFile = null;
                    try {
                        photoFile = createImageFile();
                    } catch (IOException e) {

                    }

                    if(photoFile != null) {
                        photoUri = FileProvider.getUriForFile(getApplicationContext(), getPackageName(), photoFile);
                        intent.putExtra(MediaStore.EXTRA_OUTPUT, photoUri);
                        startActivityForResult(intent, REQUEST_IMAGE_CAPTURE);
                    }
                }
            }
        });
    }

    private File createImageFile() throws IOException {
        String timeStamp = new SimpleDateFormat("yyyyMMdd_HHmmss").format(new Date());
        String imageFileName = "TEST_" + timeStamp + "_";
        File storageDir = getExternalFilesDir(Environment.DIRECTORY_PICTURES);
        File image = File.createTempFile(
            imageFileName,

```

```

        ".jpg",
        storageDir
    );

    imageFilePath = image.getAbsolutePath();
    return image;
}

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    //super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {
        Bitmap bitmap = BitmapFactory.decodeFile(imageFilePath);
        ExifInterface exif = null;

        try {
            exif = new ExifInterface(imageFilePath);
        } catch (IOException e) {
            e.printStackTrace();
        }

        int exifOrientation;
        int exifDegree;

        if (exif != null) {
            exifOrientation = exif.getAttributeInt(ExifInterface.TAG_ORIENTATION, ExifInterface.ORIENTATION_NORMAL);
            exifDegree = exifOrientationToDegrees(exifOrientation);
        } else {
            exifDegree = 0;
        }

        ((ImageView) findViewById(R.id.iv_result)).setImageBitmap(rotate(bitmap, exifDegree));
    }
}

private int exifOrientationToDegrees(int exifOrientation) {
    if (exifOrientation == ExifInterface.ORIENTATION_ROTATE_90) {
        return 90;
    } else if (exifOrientation == ExifInterface.ORIENTATION_ROTATE_180) {
        return 180;
    } else if (exifOrientation == ExifInterface.ORIENTATION_ROTATE_270) {
        return 270;
    }
    return 0;
}

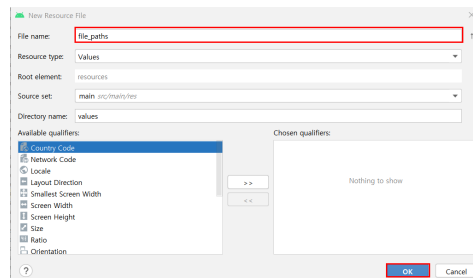
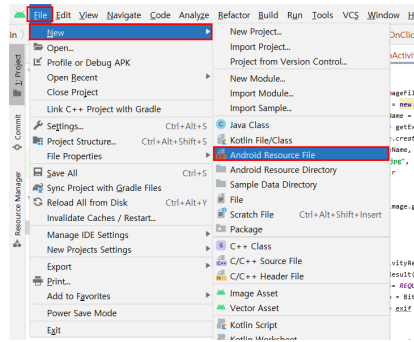
private Bitmap rotate(Bitmap bitmap, float degree) {
    Matrix matrix = new Matrix();
    matrix.postRotate(degree);
    return Bitmap.createBitmap(bitmap, 0, 0, bitmap.getWidth(), bitmap.getHeight());
}

PermissionListener permissionListener = new PermissionListener() {
    @Override
    public void onPermissionGranted() {
        Toast.makeText(getApplicationContext(), "권한이 허용됨", Toast.LENGTH_SHORT).show();
    }

    @Override
    public void onPermissionDenied(ArrayList<String> deniedPermissions) {
        Toast.makeText(getApplicationContext(), "권한이 거부됨", Toast.LENGTH_SHORT).show();
    }
};
}

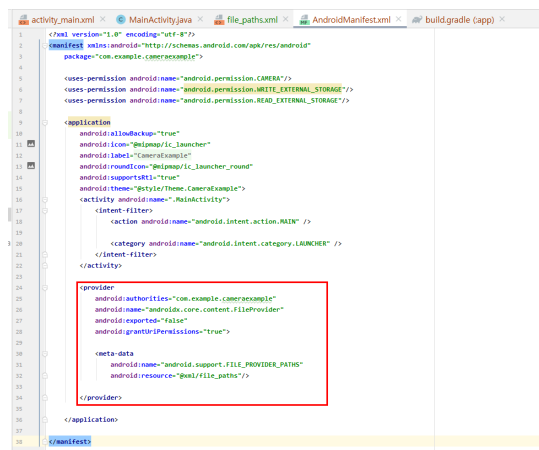
```

## 5. file\_paths.xml 추가 및 코드



```
<?xml version="1.0" encoding="utf-8"?>
<paths xmlns:android="http://schemas.android.com/apk/res/android">
  <external-path
    name="my_images"
    path="Android/data/com.example.cameraexample/files/Pictures"/>
</paths>
```

## 6. AndroidManifest.xml에서 코드 추가



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

  <uses-permission android:name="android.permission.CAMERA"/>
  <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
  <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>

  <application
```

```

        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.CameraExample">
<activity android:name=".MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

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

<provider
    android:authorities="com.example.cameraexample"
    android:name="androidx.core.content.FileProvider"
    android:exported="false"
    android:grantUriPermissions="true">

    <meta-data
        android:name="android.support.FILE_PROVIDER_PATHS"
        android:resource="@xml/file_paths"/>

</provider>
</application>
</manifest>

```