Demo31-调用摄像头拍照+选取相册的照片

1. activity\_main.xml

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

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

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent" >

<Button

android:id="@+id/take\_photo"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Take Photo" />

<Button

android:id="@+id/choose\_from\_album"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Choose From Album" />

<ImageView

android:id="@+id/picture"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal" />

</LinearLayout>

1. MainActivity.java

import android.Manifest;

import android.annotation.TargetApi;

import android.content.ContentUris;

import android.content.Intent;

import android.content.pm.PackageManager;

import android.database.Cursor;

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

import android.net.Uri;

import android.os.Build;

import android.os.Bundle;

import android.os.Environment;

import android.provider.DocumentsContract;

import android.provider.MediaStore;

import android.support.annotation.NonNull;

import android.support.v4.app.ActivityCompat;

import android.support.v4.content.ContextCompat;

import android.support.v4.content.FileProvider;

import android.support.v7.app.AppCompatActivity;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.Toast;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.IOException;

public class MainActivity extends AppCompatActivity {

public static final int TAKE\_PHOTO = 1;

public static final int CHOOSE\_PHOTO = 2;

private ImageView picture;

private Uri imageUri;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

Button takePhoto = (Button) findViewById(R.id.take\_photo);

Button chooseFromAlbum = (Button) findViewById(R.id.choose\_from\_album);

picture = (ImageView) findViewById(R.id.picture);

takePhoto.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

// 创建File对象，用于存储拍照后的图片

File outputImage = new File(getExternalCacheDir(), "output\_image.jpg");

try {

if (outputImage.exists()) {

outputImage.delete();

}

outputImage.createNewFile();

} catch (IOException e) {

e.printStackTrace();

}

if (Build.VERSION.SDK\_INT < 24) {

imageUri = Uri.fromFile(outputImage);

} else {

imageUri = FileProvider.getUriForFile(MainActivity.this, "com.example.cameraalbumtest.fileprovider", outputImage);

}

// 启动相机程序

Intent intent = new Intent("android.media.action.IMAGE\_CAPTURE");

intent.putExtra(MediaStore.EXTRA\_OUTPUT, imageUri);

startActivityForResult(intent, TAKE\_PHOTO);

}

});

chooseFromAlbum.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if (ContextCompat.checkSelfPermission(MainActivity.this, Manifest.permission.WRITE\_EXTERNAL\_STORAGE) != PackageManager.PERMISSION\_GRANTED) {

ActivityCompat.requestPermissions(MainActivity.this, new String[]{ Manifest.permission. WRITE\_EXTERNAL\_STORAGE }, 1);

} else {

openAlbum();

}

}

});

}

private void openAlbum() {

Intent intent = new Intent("android.intent.action.GET\_CONTENT");

intent.setType("image/\*");

startActivityForResult(intent, CHOOSE\_PHOTO); // 打开相册

}

@Override

public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {

switch (requestCode) {

case 1:

if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION\_GRANTED) {

openAlbum();

} else {

Toast.makeText(this, "You denied the permission", Toast.LENGTH\_SHORT).show();

}

break;

default:

}

}

@Override

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

switch (requestCode) {

case TAKE\_PHOTO:

if (resultCode == RESULT\_OK) {

try {

// 将拍摄的照片显示出来

Bitmap bitmap = BitmapFactory.decodeStream(getContentResolver().openInputStream(imageUri));

picture.setImageBitmap(bitmap);

} catch (Exception e) {

e.printStackTrace();

}

}

break;

case CHOOSE\_PHOTO:

if (resultCode == RESULT\_OK) {

// 判断手机系统版本号

if (Build.VERSION.SDK\_INT >= 19) {

// 4.4及以上系统使用这个方法处理图片

handleImageOnKitKat(data);

} else {

// 4.4以下系统使用这个方法处理图片

handleImageBeforeKitKat(data);

}

}

break;

default:

break;

}

}

@TargetApi(19)

private void handleImageOnKitKat(Intent data) {

String imagePath = null;

Uri uri = data.getData();

Log.d("TAG", "handleImageOnKitKat: uri is " + uri);

if (DocumentsContract.isDocumentUri(this, uri)) {

// 如果是document类型的Uri，则通过document id处理

String docId = DocumentsContract.getDocumentId(uri);

if("com.android.providers.media.documents".equals(uri.getAuthority())) {

String id = docId.split(":")[1]; // 解析出数字格式的id

String selection = MediaStore.Images.Media.\_ID + "=" + id;

imagePath = getImagePath(MediaStore.Images.Media.EXTERNAL\_CONTENT\_URI, selection);

} else if ("com.android.providers.downloads.documents".equals(uri.getAuthority())) {

Uri contentUri = ContentUris.withAppendedId(Uri.parse("content://downloads/public\_downloads"), Long.valueOf(docId));

imagePath = getImagePath(contentUri, null);

}

} else if ("content".equalsIgnoreCase(uri.getScheme())) {

// 如果是content类型的Uri，则使用普通方式处理

imagePath = getImagePath(uri, null);

} else if ("file".equalsIgnoreCase(uri.getScheme())) {

// 如果是file类型的Uri，直接获取图片路径即可

imagePath = uri.getPath();

}

displayImage(imagePath); // 根据图片路径显示图片

}

private void handleImageBeforeKitKat(Intent data) {

Uri uri = data.getData();

String imagePath = getImagePath(uri, null);

displayImage(imagePath);

}

private String getImagePath(Uri uri, String selection) {

String path = null;

// 通过Uri和selection来获取真实的图片路径

Cursor cursor = getContentResolver().query(uri, null, selection, null, null);

if (cursor != null) {

if (cursor.moveToFirst()) {

path = cursor.getString(cursor.getColumnIndex(MediaStore.Images.Media.DATA));

}

cursor.close();

}

return path;

}

private void displayImage(String imagePath) {

if (imagePath != null) {

Bitmap bitmap = BitmapFactory.decodeFile(imagePath);

picture.setImageBitmap(bitmap);

} else {

Toast.makeText(this, "failed to get image", Toast.LENGTH\_SHORT).show();

}

}

}

1. 在AndroidManifest.xml中注册内容提供器

<provider

android:name="android.support.v4.content.FileProvider"

android:authorities="com.example.cameraalbumtest.fileprovider"

android:exported="false"

android:grantUriPermissions="true">

<meta-data

android:name="android.support.FILE\_PROVIDER\_PATHS"

android:resource="@xml/file\_paths" />

</provider>

</application>

1. res--->New--->Directory--->xml---->New--->File--->file\_paths.xml

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

<paths xmlns:android="http://schemas.android.com/apk/res/android">

<external-path name="my\_images" path="" />

</paths>

1. 权限

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />