

توسعه برنامه های

موبایل

جلسه سیزدهم مجازی

بخش اول

سحر صادقی

## کار با دوربین در برنامه نویسی اندروید با مثال و نمونه کد

دو راه برای استفاده از دوربین در اپلیکیشن وجود دارد که آن دو راه عبارتند از:

- استفاده از برنامه ی دوربین اندروید موجود در اپلیکیشن خود.
- استفاده مستقیم از دوربین API در اندروید که در اپلیکیشن شما ارائه شده است.

شما می توانید برای آغاز اپلیکیشن دوربین نصب شده ی روی گوشی خود ، از `MediaStore.ACTION_IMAGE_CAPTURE` استفاده کنید که سینتکس آن به شکل زیر می باشد.

```
Intent intent = new Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
```

علاوه بر مورد بالا ، `Intent` های دیگری وجود دارند که توسط `MediaStore` ارائه می شوند که در زیر لیست شده اند:

`ACTION_IMAGE_CAPTURE_SECURE` : زمانی که دستگاه ایمن شده باشد، تصویر گرفته شده را به دوربین باز می گرداند.

`ACTON_VIDEO_CAPTURE` : برنامه ی `video` موجود در اندروید را فرا می خواند تا فیلم بگیرد.

`EXTRA_SCREEN_ORIENTATION` : جهت صفحه را به صورت عمودی یا منظره (screen) تنظیم می کند.

`EXTRA_FULL_SCREEN` : برای کنترل اینترفیس یوزر از `ViewImage` استفاده می شود.

`INTENT_ACTION_VIDEO_CAMERA` : برای آغاز کار دوربین به حالت ویدیو ، استفاده می شود.

`EXTRA_SIZE_LIMIT` : برای تعیین محدودیت اندازه ی تصویر یا فیلم گرفته شده استفاده می شود.

اکنون از متد `startActivityResult()` برای آغاز این اکتیویتی استفاده کرده و منتظر نتیجه بمانید. سینتکس آن در زیر ارائه شده است:

```
startActivityResult(intent, 0)
```

این متد در کلاس Activity تعریف شده است، که آن را از اکتیویتی اصلی فرا می خوانیم. روش های تعریف شده ای در کلاس Activity وجود دارند که همین کار را انجام می دهند اما زمانی استفاده می شوند که نه از کلاس activity ، بلکه از جایی دیگر فراخوانده آید. می توانید لیست آن ها را در زیر ببینید:

startActivityResult(Intent intent, int requestCode, Bundle options) یک اکتیویتی را آغاز می کند اما می تواند گروهی از گزینه های اضافه را با آن بگیرد.

startActivityFromChild(Activity child, Intent intent, int requestCode) اکتیویتی را زمانی که اکتیویتی شما زیر مجموعه ی یک اکتیویتی دیگری است، آغاز می کند.

startActivityFromChild(Activity child, Intent intent, int requestCode, Bundle options) درست مانند بالا عمل می کند اما می تواند مقادیر اضافه را به شکل یک بسته نرم افزاری (Bundle) بگیرد.

startActivityFromFragment(Fragment fragment, Intent intent, int requestCode) اکتیویتی را از بخشی آغاز می کند که اکنون در آن هستید.

startActivityFromFragment(Fragment fragment, Intent intent, int requestCode, Bundle options) تنها اکتیویتی را از fragment آغاز می کند، بلکه می تواند مقادیر اضافه را با آن بگیرد.

مهم نیست از کدام تابع برای آغاز فعالیت استفاده می کنید، همه ی آنها نتیجه را باز می گردانند. نتیجه می تواند با تحت الشعاع قرار دادن عملکرد onActivityResult به دست آید.

در اینجا مثالی را می بینید که نشان می دهد چگونه برنامه ی دوربین موجود را برای گرفتن عکس آغاز کنید و نتیجه را به شکل bitmap نمایش دهید.

برای اجرای این مثال به دستگاهی نیاز دارید که در آن دوربین ساپورت می شود.

1) برای ایجاد یک اپلیکیشن اندروید از Android studio IDE استفاده نموده و آن را با عنوان Camera تحت پکیج com.example.camera نام گذاری کنید .

2) فایل src/MainActivity.java را برای افزودن کد intent تغییر دهید تا کدهای intent برای دوربین را راه اندازی کند.

3) لی اوت فایل XML مربوط به res/layout/activity\_main.xml را تغییر دهید

4) برنامه را اجرا کنید و یک دستگاه اجرایی اندروید انتخاب کنید و برنامه را روی آن نصب کرده و نتایج را بررسی کنید.

در زیر محتوای موجود در اکتیویتی اصلی را مشاهده می کنید:

```
package com.example.sairamkrishna.myapplication; import android.Manifest; import
android.app.Activity; import android.app.AlertDialog; import android.content.Cont
ext; import android.content.DialogInterface; import android.content.Intent; import
android.content.SharedPreferences; import android.content.pm.PackageManager; impo
rt android.net.Uri; import android.os.Bundle; import android.provider.Settings; i
import android.support.v4.app.ActivityCompat; import android.support.v4.content.Con
textCompat; import android.support.v7.app.AppCompatActivity; import android.support
.v7.widget.Toolbar; public class MainActivity extends AppCompatActivity { pub
lic static final int MY_PERMISSIONS_REQUEST_CAMERA = 100; public static final S
tring ALLOW_KEY = "ALLOWED"; public static final String CAMERA_PREF = "camera_p
ref"; @Override protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState); setContentView(R.layout.activity_main);
if (ContextCompat.checkSelfPermission(this, Manifest.permission.CAMERA) != Package
Manager.PERMISSION_GRANTED) { if (getFromPref(this, ALLOW_KEY)) {
showSettingsAlert(); } else if (ContextCompat.checkSelfPermission(this,
Manifest.permission.CAMERA) != PackageManager.PERMISSION
GRANTED) { // Should we show an explanation?
if (ActivityCompat.shouldShowRequestPermissionRationale(this,
Manifest.permission.CAMERA)) { showAlert(); } else
{ // No explanation needed, we can request the permission.
ActivityCompat.requestPermissions(this, new String[]{Manifest
.permission.CAMERA}, MY_PERMISSIONS_REQUEST_CAMERA);
} } else { openCamera(); } } publi
c static void saveToPreferences(Context context, String key, Boolean allowed) {
SharedPreferences myPrefs = context.getSharedPreferences(CAMERA_PREF, Co
ntext.MODE_PRIVATE); SharedPreferences.Editor prefsEditor = myPrefs.edit();
prefsEditor.putBoolean(key, allowed); prefsEditor.commit(); }
public static Boolean getFromPref(Context context, String key) {
SharedPreferences myPrefs = context.getSharedPreferences(CAMERA_PREF, Co
ntext.MODE_PRIVATE); return (myPrefs.getBoolean(key, false)); }
private void showAlert() { AlertDialog alertDialog = new AlertD
ialog.Builder(MainActivity.this).create(); alertDialog.setTitle("Alert");
alertDialog.setMessage("App needs to access the Camera."); alertDialog
g.setButton(AlertDialog.BUTTON_NEGATIVE, "DONT ALLOW", new DialogInterfac
e.OnClickListener() { public void onClick(DialogInterface dialog, int
which) { dialog.dismiss(); finish(); }
}); alertDialog.setButton(AlertDialog.BUTTON_POSITIVE,
"ALLOW", new DialogInterface.OnClickListener() { pu
blic void onClick(DialogInterface dialog, int which) { dialog.dismi
ss(); ActivityCompat.requestPermissions(MainActivity.this,
new String[]{Manifest.permission.CAMERA}, MY_PERMISSIONS_REQUEST_CA
MERA); } }); alertDialog.show(); } pri
vate void showSettingsAlert() { AlertDialog alertDialog = new AlertDialog.Bu
ilder(MainActivity.this).create(); alertDialog.setTitle("Alert"); alert
Dialog.setMessage("App needs to access the Camera."); alertDialog.se
tButton(AlertDialog.BUTTON_NEGATIVE, "DONT ALLOW", new DialogInterface.On
ClickListener() { public void onClick(DialogInterface dialog
, int which) { dialog.dismiss(); //finish();
```

```

    });
    alertDialog.setButton(AlertDialog.BUTTON_POSITIVE, "SETTINGS",
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                dialog.dismiss();
                startInstalledAppDetailsActivity(MainActivity.this);
            }
        });
    alertDialog.show();
    @Override
    public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults) {
        switch (requestCode) {
            case MY_PERMISSIONS_REQUEST_CAMERA: {
                for (int i = 0; len = permissions.length; i < len; i++) {
                    String permission = permissions[i];
                    if (grantResults[i] == PackageManager.PERMISSION_DENIED) {
                        boolean showRationale = ActivityCompat.shouldShowRequestPermissionRationale(this, permission);
                        if (showRationale) {
                            showAlert();
                        } else if (!showRationale) {
                            // user denied flagging NEVER ASK AGAIN
                            // you can either enable some fall back, // disable features of your app // or open another dialog explaining // the app setting
                            // again the permission and directing to // the app setting
                            saveToPreferences(MainActivity.this, ALLOW_KEY, true);
                        }
                        // other 'case' lines to check for other permissions this app might request
                    }
                }
                @Override
                protected void onResume() {
                    super.onResume();
                }
                public static void startInstalledAppDetailsActivity(final Activity context) {
                    if (context == null) {
                        return;
                    }
                    final Intent i = new Intent();
                    i.setAction(Settings.ACTION_APPLICATION_DETAILS_SETTINGS);
                    i.addCategory(Intent.CATEGORY_DEFAULT);
                    i.setData(Uri.parse("package:" + context.getPackageName()));
                    i.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                    i.addFlags(Intent.FLAG_ACTIVITY_NO_HISTORY);
                    i.addFlags(Intent.FLAG_ACTIVITY_EXCLUDE_FROM_RECENTS);
                    context.startActivity(i);
                }
                private void openCamera() {
                    Intent intent = new Intent("android.media.action.IMAGE_CAPTURE");
                    startActivity(intent);
                }
            }
        }
    }

```

محتوای res/layout/activity\_main.xml file:

```

<?xml version="1.0" encoding="utf-8"?> <RelativeLayout 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:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin" tools:context=".MainActivity"> </RelativeLayout>

```

محتوای res/values/strings.xml برای یافتن ثابت های جدید:

```

<resources>
    <string name="app_name">My Application</string>
</resources>

```

محتوای پیش فرض: AndroidManifest.xml

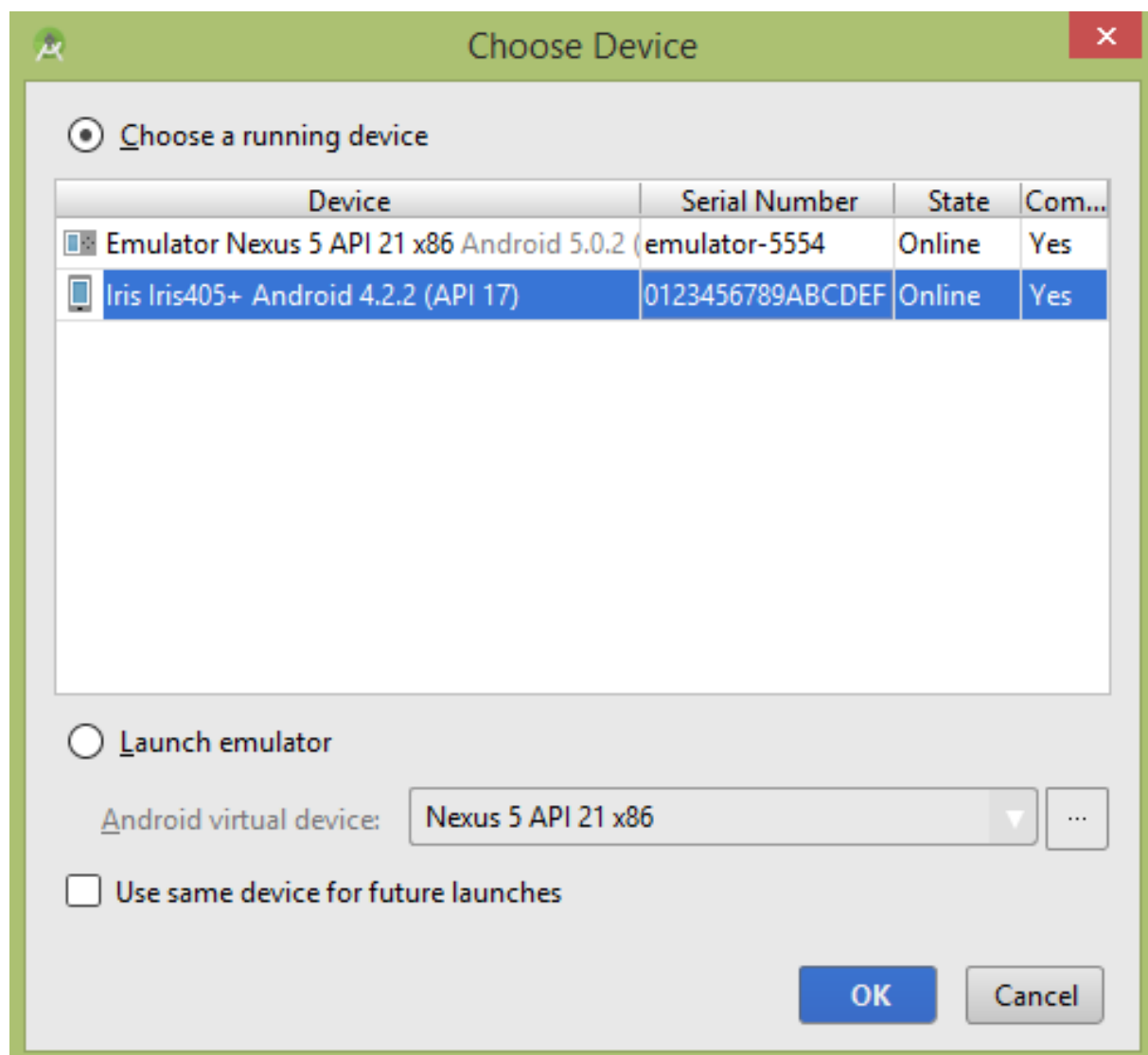
```

<?xml version="1.0" encoding="utf-8"?> <manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.sairamkrishna.myapplication" >
    <uses-permission android:name="android.permission.CAMERA" />
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.sairamkrishna.myapplication.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>

```

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

اجازه بدهید برنامه ی Camera را اجرا کنیم. فرض می کنیم که دستگاه موبایل اندروید خود را به کامپیوتر متصل کرده اید. برای اجرای برنامه از android studio ، یکی از فایل های اکتیویتی پروژه را باز کرده و روی آیکن Run از نوار ابزار کلیک کنید. قبل از شروع برنامه ی شما ، android studio پنجره ی زیر را برای انتخاب گزینه ی محل اجرای برنامه ی اندرویدتان، نمایش خواهد داد.



دستگاه موبایل خود را به عنوان یک گزینه انتخاب کنید و سپس آن را چک کنید که صفحه ی زیر را نمایش خواهد داد.

