



آموزش برنامه نویسی اندروید در محیط اندروید استودیو

کار با نوتیفیکیشن (Notification)
بخش اول: ساخت نوتیفیکیشن ساده

مدرس : سیدمهدی مطهری

www.android-studio.ir

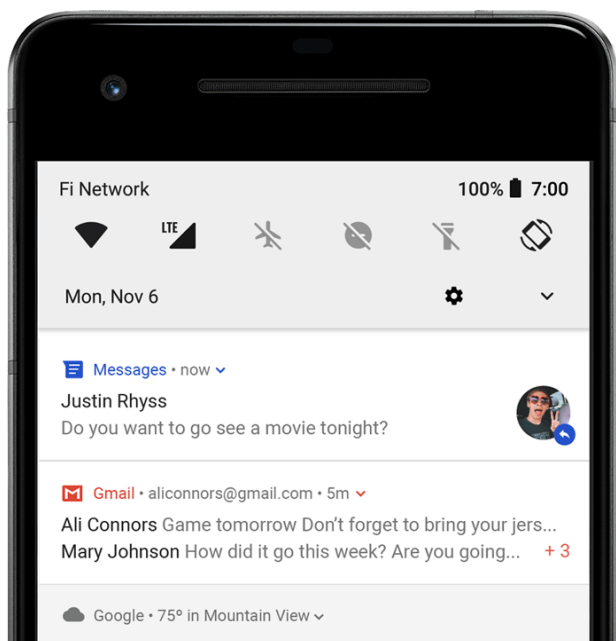


به نام خدا

نوتیفیکیشن در اندروید در حالت‌های مختلفی قابل پیاده سازی و اجراست. از یک Notification ساده شامل یک عنوان و متن و آیکون کوچک تا یک پیغام که دارای یک تصویر بزرگ، تعدادی دکمه (اکشن) و... است. در این مبحث به آموزش ساخت نوتیفیکیشن (Notification) در اندروید می‌پردازیم.

نوتیفیکیشن یا Notification چیست؟

نوتیفیکیشن یک اعلان یا پیغام است که به جهت یادآوری، هشدار یا اطلاع رسانی یک رویداد به کاربر نمایش داده می‌شود. این اعلان ابتدا در قالب یک آیکون در قسمت نوار وضعیت یا همان Status Bar ظاهر می‌گردد که کاربر با کشیدن نوار به پایین و باز شدن Notification Drawer، نوتیفیکیشن(های) ساخته شده توسط اپلیکیشن‌ها را مشاهده می‌کند.



قطعا همه ما روزانه با تعداد زیادی از این اعلان‌ها در دیوایس اندرویدی خود سروکار داریم. Notification ها در محیط خارج از رابط کاربری برنامه ساخته می‌شوند. بنابراین این اعلان‌ها می‌توانند در هر زمانی ساخته شوند. حتی هنگامی که برنامه در حال اجرا نیست.



به عنوان مثال شما در حال تماشای یک ویدئو هستید و یک اعلان مشاهده می‌کنید که به شما اطلاع می‌دهد یک SMS دریافت شده. یا در حال وب گردی هستید و آیکون Instagram را در نوار وضعیت مشاهده می‌کنید که با کشیدن نوار به پایین، پیغامی مبنی بر درخواست فالو (Follow) از جانب یک شخص نشان داده می‌شود. یا اعلان‌های تبلیغاتی که کاربر را به خرید یک محصول یا ارتقاء برنامه از نسخه رایگان به نسخه تجاری ترغیب می‌کند.

بنابراین نوتیفیکیشن در اندروید و سایر سیستم عامل‌ها کاربرد بسیار گسترده‌ای داشته و بسته به هدف و نیاز، در فرمت‌ها و قالب‌های متعددی می‌توان آنها را اجرا کرد. این پیغام‌ها حتی هنگامی که صفحه در حالت قفل (Lock screen) قرار دارد نیز امکان نمایش اطلاعات را دارند؛ مانند اعلان یک تقویم که تاریخ شمسی، میلادی و قمری را به ما نشان می‌دهد.

انواع نوتیفیکیشن‌ها در اندروید

نوتیفیکیشن در اندروید را به روش‌ها و حالت‌های مختلفی می‌توان پیاده سازی کرد. از یک اعلان ساده شامل عنوان و توضیحات گرفته تا یک اعلان پیچیده شامل تصویر، اکشن، استفاده از Style و

همچنین توسط PendingIntent می‌توانیم تعیین کنیم کاربر پس از لمس اعلان به کدام اکتیویتی از برنامه منتقل شود.

البته در این آموزش صرفاً نحوه ساخت یک نوتیفیکیشن ساده را بررسی می‌کنیم. مباحث پیشرفته را در آموزش‌های بعد ارائه خواهم کرد.

یک پروژه جدید در **اندروید استودیو** با نام Notification می‌سازم که اکتیویتی پیش فرض آن از نوع Empty Activity است.

در این مبحث من وارد جزئیات نحوه نمایش یک اعلان نمی‌شوم و نوتیفیکیشن‌ها را صرفاً توسط Button اجرا می‌کنم.

نوتیفیکیشن ساده (Simple Notification):

در قدم اول ساده ترین نوع از نوتیفیکیشن در اندروید را پیاده سازی می‌کنم تا با کلیت کار آشنا شوید. ابتدا یک Button به activity_main.xml اضافه می‌کنم. برای مدیریت ساده‌تر دکمه‌ها لایه را به LinearLayout تغییر داده‌ام:



```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    tools:context=".MainActivity">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/simple_notif"
        android:text="Simple Notification"/>

</LinearLayout>
```

نکته: پس از درخواست چند نفر از کاربران، تم اندروید استودیو خود را از حالت تیره به روشن تغییر دادم تا افرادی که مایل به پرینت آموزشها هستند با مشکل مواجه نشوند.

در ادامه کار، کدهایی که در فایل جاوای اکتیویتی نوشته‌ام را تشریح می‌کنم:



:MainActivity.java

```
package ir.android_studio.notification;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;
import android.app.NotificationManager;
import android.content.Context;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    Button simpleButton;
    NotificationManager notifManager;

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

        simpleButton = findViewById(R.id.simple_notif);

        notifManager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);

        simpleButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                simpleNotification();
            }
        });
    }

    public void simpleNotification() {

        NotificationCompat.Builder sNotifBuilder = new NotificationCompat.Builder(this)
            .setSmallIcon(R.drawable.notif_icon)
            .setContentTitle("بروزرسانی")
            .setContentText("یک نسخه جدید از برنامه آماده دریافت است");

        notifManager.notify(1, sNotifBuilder.build());
    }
}
```

داخل بدنه اصلی کلاس MainActivity یک Button با نام simpleButton و یک آجکت از متد NotificationManager با نام notifManager جهت مدیریت نوتیفیکیشن ساخته‌ام. سپس این دو را درون متد onCreate کلاس تعریف کرده‌ام.

برای مرتب بودن کدها، کد مربوط به نوتیفیکیشن ساده را درون یک تابع جداگانه در بدنه اصلی کلاس تعریف کردم. من نام این متد را simpleNotification انتخاب کرده‌ام. برای ساخت محتوای نوتیفیکیشن لازم است تا از متد NotificationCompat.Builder یک آجکت (شیء) بسازیم. من نام این شیء را



sNotifBuilder تعیین کردم. پارامتر ورودی این متد، Context است که من this وارد نموده‌ام. سپس سه مورد از محتوای مدنظرم را به این آبجکت اضافه کرده‌ام:

- setSmallIcon: آیکون نوتیفیکیشن توسط این متد تعیین می‌شود. من از آیکون‌های موجود در خود اندروید استودیو برای تست این ویژگی استفاده کردم. جهت استفاده از این آیکون‌ها روی پوشه drawable پروژه اندروید خود راست کلیک کرده، New > Vector Asset را انتخاب کنید. در صفحه‌ی باز شده یک آیکون انتخاب کرده و تایید کنید تا به drawable افزوده شود. یا از آیکون‌های خارج از اندروید استودیو استفاده کنید که کفایت فایل را کپی کرده و روی این پوشه paste کنید. من آیکون را با نام notif_icon به پروژه اضافه کردم و همانطور که مشاهده می‌کنید همین نام در ورودی setSmallIcon تعریف شده.

نکته: با توجه به کیفیت بالای صفحه نمایش دیوایس‌های موبایلی جدید، استفاده از آیکون‌های jpg و png خروجی با کیفیت پایین را ارائه می‌دهد. بنابراین سعی کنید از آیکون‌های بُرداری (که برای اندروید عموماً xml هستند) استفاده نمائید.

نکته: setSmallIcon تنها متدی ست که تعریف آن برای یک نوتیفیکیشن ضروری است.

- setTitle: از این متد برای تعیین عنوان پیغام استفاده می‌شود.
- setText: همانطور که از نام آن پیداست برای نمایش متن پیغام بکار می‌رود.

فعلاً به همین سه مورد اکتفا می‌کنم.

در خط آخر هم توسط متد notify دو ورودی به شیء ساخته شده از NotificationManager یعنی notifManager ارسال شده است. ورودی اول یک عدد صحیح است که در اینجا ۱ را انتخاب کردم. در ورودی دوم نیز سازنده‌ی نوتیفیکیشن یعنی sNotifBuilder را build می‌کنیم.

نکته: ورودی اول به عنوان ID یا شناسه هر نوتیفیکیشن در نظر گرفته می‌شود. این شناسه می‌بایست برای هر نوتیفیکیشن منحصر بفرد باشد. در صورت مشابه بودن شناسه برای دو یا چند نوتیفیکیشن، آخرین اعلانی که اجرا شده جایگزین اعلان قبلی با شناسه برابر خواهد شد.



این یک نوتیفیکیشن بسیار ساده است.

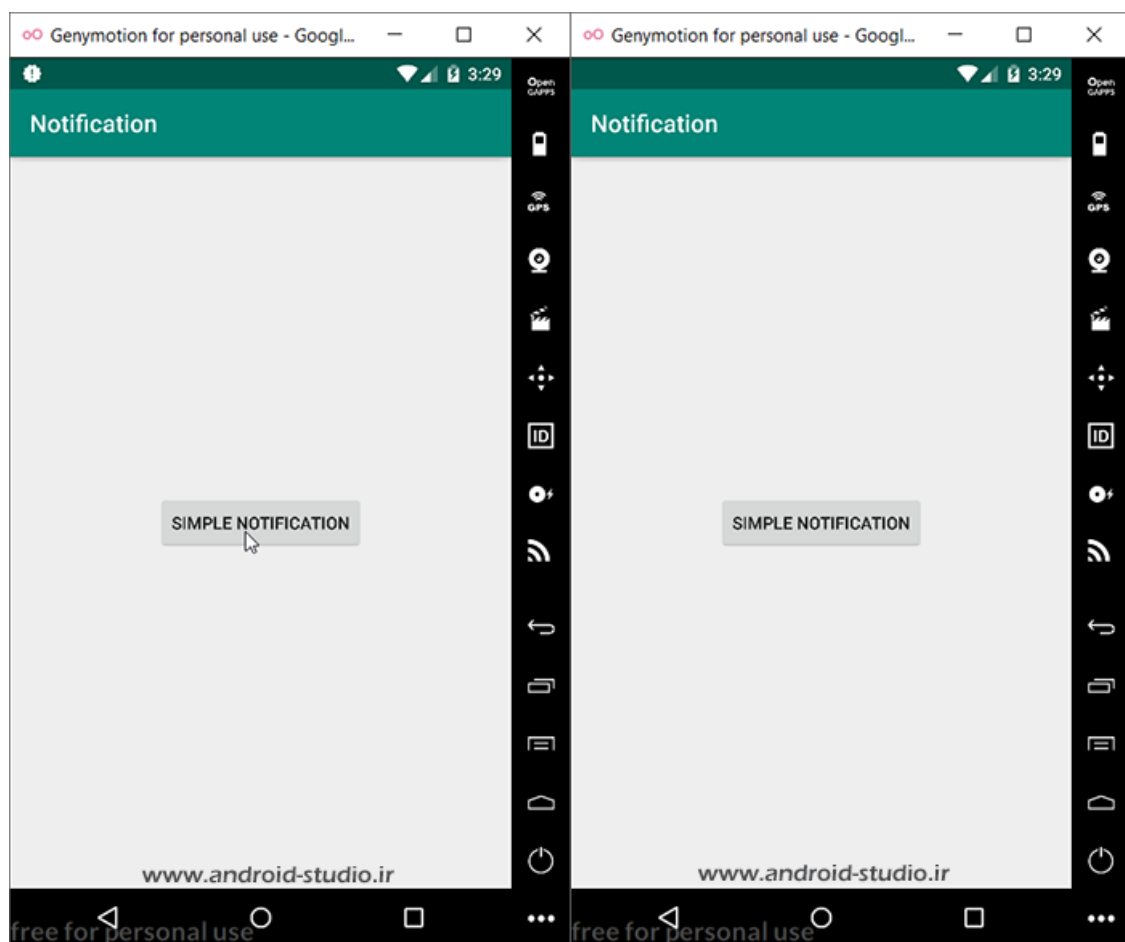
```
public void simpleNotification() {
    NotificationCompat.Builder sNotifBuilder = new NotificationCompat.Builder( context: this)
        .setSmallIcon(R.drawable.notif_icon)
        .setContentTitle("بروزرسانی")
        .setContentText("یک نسخه جدید از برنامه آماده دریافت است");
    notifManager.notify( id: 1, sNotifBuilder.build());
}
```

Builder(android.content.Context) is deprecated more... (Ctrl+F1)

www.android-studio.ir

در تصویر فوق مشاهده می‌کنید برای NotificationCompat.Builder گفته شده این Builder منقضی (deprecated) شده است. دلیل آن را در ادامه متن متوجه خواهید شد.

پروژه را روی **شبه ساز اندروید** با API پایین‌تر از ۲۶ یعنی (Android 8 (Oreo اجرا می‌کنم. (اگر به دیوایس مجازی یا حقیقی پایین‌تر از API 26 دسترسی ندارید از این اجرا صرف نظر کرده و ادامه مبحث را دنبال کنید).





نوتیفیکیشن به درستی اجرا شد. با کلیک روی دکمه، آیکون در نوار وضعیت ظاهر شده و با کشیدن آن به پایین، محتوایی که تعریف کرده بودم درون نوتیفیکیشن قرار دارد. این پیغام با کشیدن به سمت چپ یا راست حذف می‌شود.

حالا پروژه را روی یک دیوایس با API 26 و یا بالاتر اجرا می‌کنم:



با کلیک روی دکمه، نوتیفیکیشن اجرا نشده و ارور زیر ظاهر می‌شود:

Developer warning for package "ir.android_studio.notification"

Failed to post notification on channel "null"

See log for more details

با این مضمون که امکان ارسال نوتیفیکیشن روی channel با مقدار null وجود ندارد. اما علت چیست:

اضافه شدن Notification Channel در Android Oreo

در Android 8 یا همان Oreo نوتیفیکیشن در اندروید دچار تغییراتی شد. در این نسخه و نسخه‌های جدیدتر، نوتیفیکیشن‌ها باید در قالب کانال‌هایی تقسیم بندی شوند. این قابلیت به جهت کنترل بیشتر کاربر بر روی اعلان‌های دریافتی اضافه شده.

به عنوان مثال یک اپلیکیشن فروشگاه‌های را در نظر بگیرید. نوتیفیکیشن‌های این برنامه شامل محصولات جدید و همچنین تخفیفات روزانه است. اما یک کاربر مایل به دریافت پیغام‌های مربوط به محصولات جدید نیست و تنها می‌خواهد اعلان‌های تخفیف روزانه را دریافت نماید. حالا با اضافه شدن این قابلیت،



صاحبان دیوایس‌های Android O و به بالا به راحتی می‌توانند تعیین کنند کدام دسته از نوتیفیکیشن‌های هر برنامه نمایش داده شود.

یک Channel ساخته و آنرا به نوتیفیکیشن معرفی می‌کنم:

```
package ir.android_studio.notification;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.content.Context;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    Button simpleButton;
    NotificationManager notifManager;
    String offerChannelId = "Offers";

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

        simpleButton = findViewById(R.id.simple_notif);

        notifManager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);

        simpleButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                createNotifChannel();
                simpleNotification();

            }
        });

        private void createNotifChannel() {

            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {

                String offerChannelName = "Shop offers";
                String offerChannelDescription= "Best offers for customers";
                int offerChannelImportance = NotificationManager.IMPORTANCE_DEFAULT;

                NotificationChannel notifChannel = new NotificationChannel(offerChannelId, offerChannelName,
offerChannelImportance);
                notifChannel.setDescription(offerChannelDescription);

                notifManager.createNotificationChannel(notifChannel);

            }

        }

        public void simpleNotification() {

            NotificationCompat.Builder sNotifBuilder = new NotificationCompat.Builder(this, offerChannelId)
```



```

        .setSmallIcon(R.drawable.notif_icon)
        .setContentTitle("بروزرسانی")
        .setContentText("یک نسخه جدید از برنامه آماده دریافت است");

        notifManager.notify(1, sNotifBuilder.build());
    }
}

```

هر کانال باید یک شناسه (id) و نام منحصر بفرد داشته باشد. برای تعیین شناسه کانال یک String با نام offerChannelId به کلاس اضافه کردم که مقدار آن Offers است. برای ساخت NotificationChannel یک تابع با نام دلخواه createNotifChannel به کلاس اضافه کردم. گفتیم کانال فقط در اندروید Oreo و به بالا ضروری است. بنابراین لزومی ندارد کانال در اندرویدهای پایین‌تر از این نسخه هم ساخته شوند. برای اینکار ابتدا یک if تعریف می‌کنم:

```

if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
    //Channel codes...
}

```

با استفاده از این شرط، برنامه ما سیستم عامل دیوایس کاربر را چک کرده و تنها در صورتی که درون بلاک را اجرا می‌کند که سیستم عامل بزرگتر یا مساوی (\geq) نسخه O یعنی Oreo باشد.

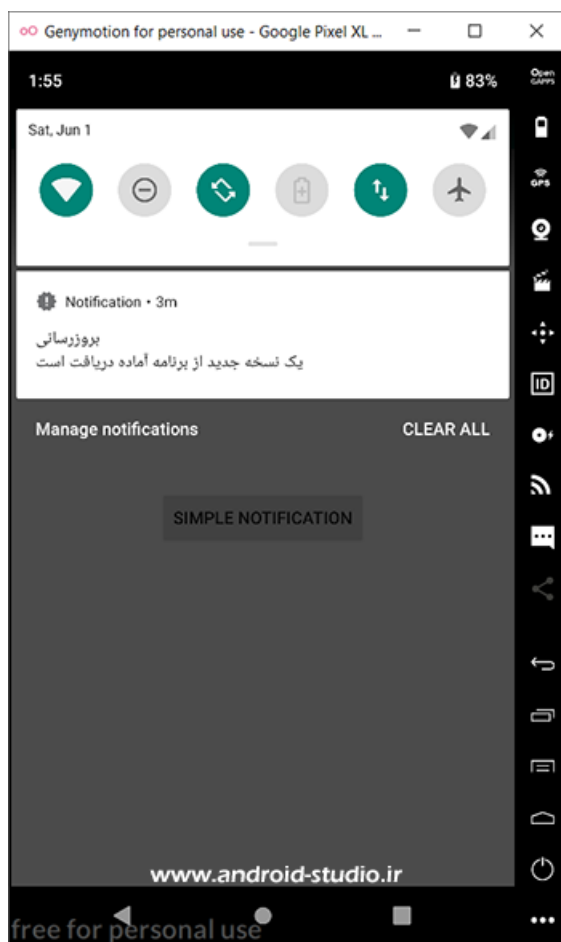
داخل این بلاک یک نام (offerChannelName) و توضیح (offerChannelDescription) برای کانال تعریف شده. در خط یک متغیر از نوع int و با نام offerChannelImportance اضافه کردم. این ویژگی همانطور که از نام آن (importance) پیداست برای تعیین درجه اهمیت نوتیفیکیشن استفاده می‌شود. من مقدار پیش‌فرض یعنی IMPORTANCE_DEFAULT را انتخاب کردم. در ادامه مبحث با سایر حالت‌ها نیز آشنا خواهید شد.

سپس یک آبجکت با نام notifChannel از متد NotificationChannel ساختم. در قسمت سازنده متد سه ورودی تعریف شده: شناسه کانال (offerChannelId)، نام (offerChannelName) و درجه اهمیت پیغام یعنی offerChannelImportance.

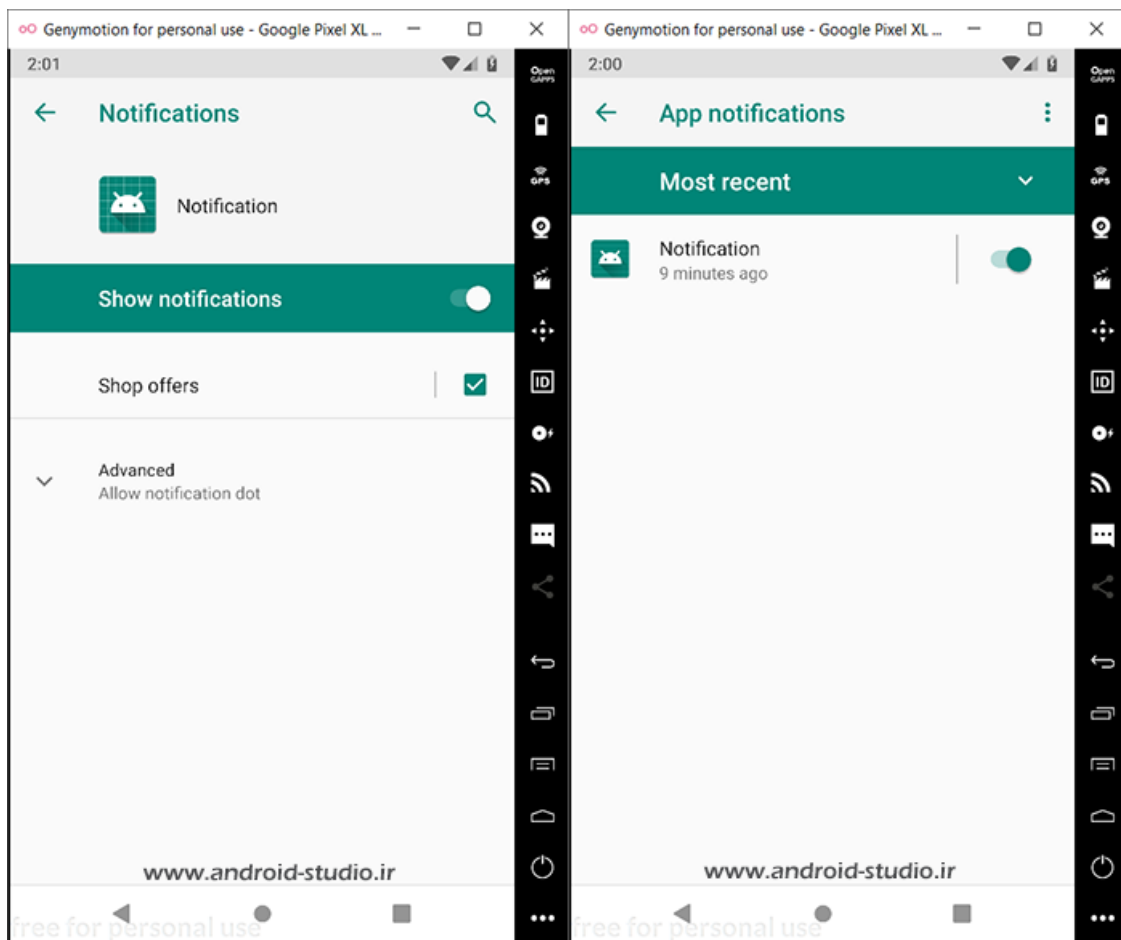
در خط بعد شرح کانال (offerChannelDescription) را به کانال اضافه کرده‌ام.

در نهایت notifChannel را توسط متد createNotificationChannel ساخته و به notifManager می‌فرستم. همچنین createNotifChannel را نیز به onClick دکمه اضافه می‌کنم.

حالا مجدد پروژه را روی دیوایس API 26 یا بالاتر اجرا می‌کنم:



نوتیفیکیشن با موفقیت ساخته شد.
روی Manage notifications کلیک می‌کنم:



در مرحله اول نام برنامه من در لیست نمایش داده شده که یک دکمه برای فعال یا غیرفعال کردن نوتیفیکیشن‌های این برنامه مقابل آن قرار دارد. با خاموش کردن این دکمه، تمامی نوتیفیکیشن‌های این برنامه غیر فعال خواهد شد. اما با کلیک روی نام برنامه، لیست کانال‌های موجود نمایش داده می‌شود. همانطور که قبلاً اشاره شد، کاربر در اینجا می‌تواند تعیین کند کدام دسته از اعلان‌ها را دریافت کند.

ویژگی‌های Importance و Priority

در قسمت قبل یک importance با درجه اهمیت IMPORTANCE_DEFAULT برای کانال تعیین کردیم. این سطح اهمیت در ۴ سطح قابل تعریف است:

IMPORTANCE_MIN (پایین‌ترین سطح)، IMPORTANCE_LOW (سطح متوسط)، IMPORTANCE_DEFAULT (سطح بالا یا پیش‌فرض) و IMPORTANCE_HIGH که بالاترین سطح اهمیت اعلان (سطح اضطراری) را نشان می‌دهد.



توسط این ویژگی‌ها می‌توانیم تعیین کنیم هرکدام از نوتیفیکیشن‌های برنامه ما چقدر اهمیت داشته و نمایش آن به کاربر چقدر ضرورت دارد. هرکدام از این ۴ سطح خصوصیات متفاوتی با سایر آیتم‌ها دارد. به عنوان مثال اگر درجه را از نوع `IMPORTANCE_LOW` تعیین کنیم، آیکون نوتیفیکیشن در نوار وضعیت نشان داده نشده و علاوه بر آن، صدایی هم پخش نخواهد شد. صرفاً کاربر آنرا در `Notification Drawer` مشاهده خواهد کرد. ولی در سطح `IMPORTANCE_DEFAULT` که در قسمت قبل تست کردیم، آیکون در نوار وضعیت اضافه شده و صدای دریافت نوتیفیکیشن نیز برای کاربر پخش می‌شود. جدول زیر به خوبی درجه‌ها را مقایسه می‌کند:

User-visible importance level	Importance (Android 8.0 and higher)	Priority (Android 7.1 and lower)
Urgent Makes a sound and appears as a heads-up notification	<code>IMPORTANCE_HIGH</code>	<code>PRIORITY_HIGH</code> or <code>PRIORITY_MAX</code>
High Makes a sound	<code>IMPORTANCE_DEFAULT</code>	<code>PRIORITY_DEFAULT</code>
Medium No sound	<code>IMPORTANCE_LOW</code>	<code>PRIORITY_LOW</code>
Low No sound and does not appear in the status bar	<code>IMPORTANCE_MIN</code>	<code>PRIORITY_MIN</code>

طبق این جدول، نوتیفیکیشنی که در درجه `IMPORTANCE_HIGH` قرار گرفته باشد در `Notification Drawer` بالای سایر اعلان‌ها قرار می‌گیرد.

اما این سطوح فقط در دیوایس‌های `Oreo` و به بعد تاثیرگذار هستند زیرا این ویژگی درون کانال تعریف می‌شود. اندروید ویژگی مشابهی با عنوان `Priority` معرفی کرده که همان درجه بندی را تعیین می‌کند اما برای دیوایس‌های قبلاً از `Oreo` یعنی `API 25` و کمتر. البته برخلاف `importance` که تعیین آن برای کانال ضروریست، استفاده از `priority` ضرورتی ندارد و به دلخواه توسعه دهنده می‌تواند اضافه شود. این ویژگی‌ها توسط متد `setPriority` به نوتیفیکیشن افزوده می‌شوند. مانند مثال زیر:

```
NotificationCompat.Builder sNotifBuilder = new NotificationCompat.Builder(this, offerChannelId)
    .setSmallIcon(R.drawable.notif_icon)
    .setContentTitle("بروزرسانی")
    .setContentText("یک نسخه جدید از برنامه آماده دریافت است")
    .setPriority(NotificationCompat.PRIORITY_DEFAULT);
```

در ادامه چند ویژگی دیگر که برای `NotificationCompat.Builder` قابل استفاده هستند را معرفی می‌کنم:



```
NotificationCompat.Builder sNotifBuilder = new NotificationCompat.Builder(this, offerChannelId)
    .setSmallIcon(R.drawable.notif_icon)
    .setContentTitle("بروزرسانی")
    .setContentText("یک نسخه جدید از برنامه آماده دریافت است")
    .setVibrate(new long[]{100,500,500,500,500,500})
    .setLargeIcon(BitmapFactory.decodeResource(getResources(), R.drawable.android))
    .setSound(RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION))
    .setVisibility(NotificationCompat.VISIBILITY_PUBLIC);
```

setVibrate: اولین موردی که به نوتیفیکیشن اضافه شده **setVibrate** است. توسط این متد هنگام اجرای نوتیفیکیشن، لرزش یا ویبره دیوایس فعال می‌شود. نحوه اجرای ویبره نیز قابل تنظیم است:

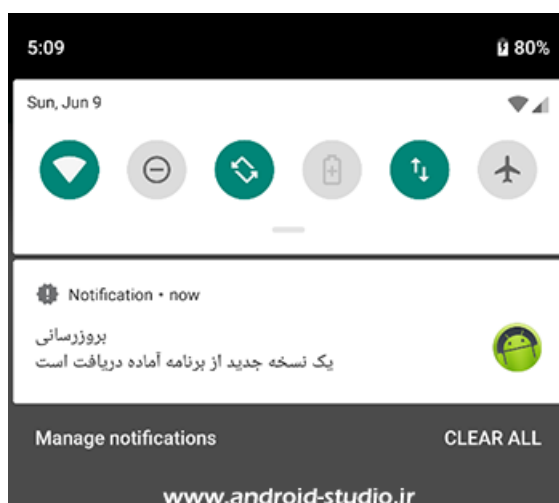
```
.setVibrate(new long[]{100,500,500,500,500,500})
```

اعداد نشان دهنده زمان و بر حسب میلی ثانیه بوده و فرمت قرارگیری به صورت زیر می‌باشد:

```
.setVibrate(new long[]{delay,play,sleep,play,sleep,...})
```

تایم اول (۱۰۰) مربوط به مدت زمان تاخیر اجرای ویبره از هنگام اجرای نوتیفیکیشن است. مابقی اعداد به ترتیب مدت زمان اجرای ویبره و توقف آنرا تعیین می‌کنند و محدودیتی در تعداد این اجرا و توقف وجود ندارد. در کد فوق بعد از ۱۰۰ میلی ثانیه تاخیر از زمان اجرا نوتیفیکیشن، ۳ بار به مدت ۵۰۰ میلی ثانیه ویبره اجرا شده و بعد از هر اجرا نیز ۵۰۰ میلی ثانیه توقف صورت می‌گیرد.

setLargeIcon: با استفاده از این متد می‌توان یک آیکون و تصویر بزرگ (علاوه بر آیکون اصلی نوتیفیکیشن) به کنار پیغام اضافه کرد:



setSound: فارغ از اینکه importance کانال یا priority نوتیفیکیشن از چه نوعی انتخاب شده باشد، با استفاده از این متد صدای نوتیفیکیشن فعال می‌گردد.



setVisibility: برای این متد سه حالت مختلف تعریف شده:

- **VISIBILITY_PUBLIC:** در این حالت نوتیفیکیشن به طور کامل بر روی صفحه قفل دیوایس (Lock screen) ظاهر شده و آیکون، عنوان و متن اعلان قابل مشاهده است.
- **VISIBILITY_PRIVATE:** در این حالت نوتیفیکیشن در لاک اسکرین ظاهر شده اما عنوان و متن اعلان مخفی است تا زمانی که قفل دیوایس باز شود.
- **VISIBILITY_SECRET:** در این حالت نوتیفیکیشن در صفحه قفل کاملاً مخفی بوده و مشاهده نمی‌شود.

متدهای متعدد دیگری برای مدیریت نوتیفیکیشن‌ها وجود دارد که در صورت تمایل و با مطالعه سایر منابع (از جمله لینک‌های معرفی شده در انتهای همین مطلب) می‌توانید از آنها نیز استفاده کنید.

برای notificationChannel هم ویژگی‌های مختلفی تعریف شده. به کد زیر دقت کنید:

```
NotificationChannel notifChannel = new NotificationChannel(offerChannelId, offerChannelName,
offerChannelImportance);
notifChannel.setDescription(offerChannelDescription);
notifChannel.enableVibration(true);
notifChannel.enableLights(true);
notifChannel.setLightColor(Color.GREEN);
```

قبلاً setDescription را برای نمایش توضیحات کانال استفاده کرده بودم. سه متد دیگر را در ادامه معرفی می‌کنم:

enableVibration: برای فعال کردن ویبره استفاده می‌شود. مانند setVibrate در NotificationCompat با این تفاوت که امکان شخصی سازی و تعیین زمان تاخیر، اجرا و توقف وجود ندارد. اینکه از کدام روش برای فعال شدن ویبره استفاده می‌کنید به سلیقه شما بستگی دارد. و البته یادآوری این نکته که enableVibration فقط از API 26 به بالا اجرا خواهد شد.

enableLights: اضافه کردن این متد با مقدار true باعث می‌شود چراغ اعلان (LED) موجود در بالای صفحه نمایش (دیوایس برای این کانال فعال شود).

setLightColor: رنگ LED هنگام اجرای نوتیفیکیشن توسط این متد تعیین می‌شود. در این مثال من رنگ سبز را انتخاب کرده‌ام.



کد کامل MainActivity.java:

```
package ir.android_studio.notification;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.content.Context;
import android.graphics.BitmapFactory;
import android.graphics.Color;
import android.media.RingtoneManager;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    Button simpleButton;
    NotificationManager notifManager;
    String offerChannelId = "Offers";

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

        simpleButton = findViewById(R.id.simple_notif);

        notifManager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);

        simpleButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                createNotifChannel();
                simpleNotification();

            }
        });
    }

    private void createNotifChannel() {

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {

            String offerChannelName = "Shop offers";
            String offerChannelDescription= "Best offers for customers";
            int offerChannelImportance = NotificationManager.IMPORTANCE_DEFAULT;

            NotificationChannel notifChannel = new NotificationChannel(offerChannelId, offerChannelName,
offerChannelImportance);
            notifChannel.setDescription(offerChannelDescription);
            //notifChannel.enableVibration(true);
            notifChannel.enableLights(true);
            notifChannel.setLightColor(Color.GREEN);

            notifManager.createNotificationChannel(notifChannel);

        }
    }

    public void simpleNotification() {

        NotificationCompat.Builder sNotifBuilder = new NotificationCompat.Builder(this, offerChannelId)
            .setSmallIcon(R.drawable.notif_icon)
    }
}
```



```

        .setContentType("بروزرسانی")
        .setContentText("یک نسخه جدید از برنامه آماده دریافت است")
        .setVibrate(new long[]{100,500,500,500,500,500})
        .setLargeIcon(BitmapFactory.decodeResource(getResources(), R.drawable.android))
        .setSound(RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION))
        .setVisibility(NotificationCompat.VISIBILITY_PUBLIC);

        notifManager.notify(1, sNotifBuilder.build());
    }
}

```

در این مبحث با نحوه ساخت یک Notification ساده جهت نمایش نوتیفیکیشن در اندروید و همچنین استفاده از notificationChannel برای سازگاری اعلان‌ها با Android O و بالاتر آشنا شدیم.

در مبحث آینده به ساخت نوتیفیکیشن‌های پیشرفته می‌پردازیم. از جمله افزودن اکشن (Action) به اعلان، استفاده از Style، مدیریت Notification badge، ساخت Notification Groups، حالت‌های مختلف نوتیفیکیشن مانند Inbox Style، BigPicture Style، Messaging Style و...

مطالعه‌ی بیشتر:

<https://developer.android.com/guide/topics/ui/notifiers/notifications>

<https://developer.android.com/training/notify-user/build-notification>

<https://developer.android.com/training/notify-user/channels>

توجه: سورس پروژه اندروید درون پوشه Exercises قرار دارد

با ارائه انتقادات و پیشنهادات خود، ما را در ارائه آموزش‌های بهتر یاری فرمائید.

www.android-studio.ir