آموزش برنامه نویسی اندروید
در محیط اندروید استودیو

کار با ویبره (Vibrate)

مدرس : سیدمهدی مطهری

www.android-studio.ir

به نام خدا

در این جلسه از سری مباحث آموزش برنامه نویسی اندروید به نحوه کار با ویبره (Vibrate) در اندروید، بررسی انواع حالت‌های ویبره و همچنین تغییرات نحوه پیاده سازی ویبره در نسخه‌های جدید سیستم عامل اندروید می‌پردازیم.

قابلیت ویبره یا Vibrate در اندروید

قطعا این قابلیت نیازی به توضیح ندارد زیرا هرکدام از ما روزانه ده‌ها بار لرزش موبایل، تبلت یا ساعت هوشمند خود را احساس می‌کنیم. ویبره یا لرزش دستگاه کاربردهای متفاوتی دارد که برای مثال می‌توان به هشدارها (ثبت اطلاعات نادرست در یک فیلد و...)، اطلاع رسانی‌ها (مانند دریافت پیامک و تماس) و ارائه تعامل بهتر کاربر با گیم‌ها و برنامه‌های سرگرمی اشاره کرد.

در ادامه آموزش با ساخت یک پروژه اندرویدی، به نحوه پیاده سازی Vibrate و انواع حالت‌های آن می‌پردازم.

ساخت پروژه Vibrate (ویبره) در اندروید استودیو

مطابق مبحث آموزش ساخت پروژه در اندروید استودیو یک پروژه اندرویدی با نام Vibrator می‌سازم. اکتیویتی را از نوع Empty Activity و زبان را Java انتخاب کردم.

جزئیات و سازگاری با نسخه‌های مختلف را قدم به قدم و در قالب چند دکمه بررسی می‌کنیم. قبل از هرکار باید مجوز دسترسی برنامه به قابلیت ویبره را در مانیفست پروژه تعریف کنیم؛ در غیر اینصورت درخواست‌های فعال و غیر فعال کردن سخت افزار Vibrator توسط سیستم عامل تایید نخواهد شد.

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

با تعریف مجوز فوق در تگ اصلی manifest دسترسی برنامه به ویبره دستگاه فراهم می‌شود. این دسترسی جزء مجوزهای خطرناک (Dangerous Permissions) نیست بنابراین نیازی به تعریف آن در Runtime Permission نخواهد بود.

AndroidManifest.xml

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

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

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

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

</manifest>
```

ویبره معمولی

ساده ترین حالت لرزش دستگاه، یک لرزش ممتد و کوتاه است. ابتدا یک Button در layout اکتیویتی تعریف می‌کنم:

activity_main.xml

```
<?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"
    tools:context=".MainActivity">

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="ویبره ساده"
        android:id="@+id/simple_vibrator" />

</LinearLayout>
```

سپس اکتیویتی را به صورت زیر تکمیل می‌کنم:

MainActivity.java

```
package ir.android_studio.vibrator;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.os.Bundle;
import android.os.Vibrator;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    Button simpleButton;
    Vibrator mVibrator;

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

```
        mVibrator = (Vibrator) getSystemService(Context.VIBRATOR_SERVICE);

        simpleButton = findViewById(R.id.simple_vibrator);
        simpleButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                mVibrator.vibrate(300);

            }
        });

    }
}
```

ابتدا یک نمونه از کلاس Vibrator با نام دلخواه mVibrator ایجاد و سپس API ویبره را درون متد onCreate() تعریف کردم:

```
mVibrator = (Vibrator) getSystemService(Context.VIBRATOR_SERVICE);
```

در ادامه یک شنونده (Listener) برای دکمه تعریف شده که درون آن متد vibrate() فراخوانی شده است. این متد در حالت ساده یک ورودی از جنس long می‌گیرد که مدت زمان لرزش دستگاه بر حسب میلی ثانیه را تعیین می‌کند. در اینجا من عدد ۳۰۰ را انتخاب کردم بنابراین مدت زمان لرزش ۰.۳ ثانیه خواهد بود.

اما تعریف ویبره به اینصورت از API 26 (Android Oreo) به بعد منقضی (Deprecated) شده بنابراین برای دیوایس‌های اندرویدی نسخه O و به بالا، متد را به صورت زیر بازنویسی می‌کنم:

```
public void onClick(View view) {

    if (Build.VERSION.SDK_INT >= 26) {

        mVibrator.vibrate(VibrationEffect.createOneShot(300,
VibrationEffect.DEFAULT_AMPLITUDE));

    } else {

        mVibrator.vibrate(300);

    }

}
```

VibrationEffect.createOneShot یک لرزش ممتد ایجاد می‌کند.

نکته: علیرغم منقضی شدن vibrate(long milliseconds) با اینحال باز هم در دیوایس‌های API 26 و به بالا به درستی عمل می‌کند.

ویبره الگو دار

در API ویبره سیستم عامل اندروید، علاوه بر ویبره ساده و ممتد امکان پیاده سازی ویبره‌های با الگوی معین نیز وجود دارد. یک متغیر از جنس long[] به صورت زیر در بدنه کلاس اکتیویتی تعریف می‌کنم:

```
long[] vibratePattern = {0,200,400,200,300,200,500,200,500,200,200,1000};
```

متغیر فوق شامل آرایه‌ای از اعداد است که بر حسب میلی ثانیه تعریف شده. اعداد به ترتیب، مدت زمان مکث و لرزش را تعیین می‌کنند. این آرایه صرفا یک مثال است و محدودیتی در تعداد اندیس‌های آن وجود ندارد. اندیس‌ها را جابجا و کم و زیاد کنید یا اعداد را ویرایش کرده تا نحوه عملکرد آن را به خوبی درک کنید.

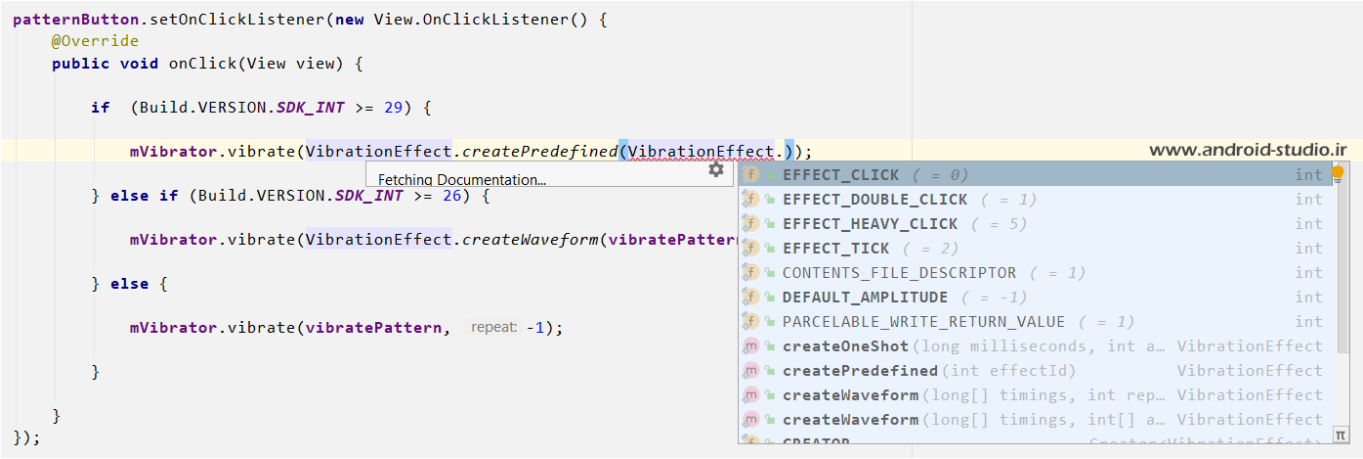
یک دکمه دیگر به layout اضافه کرده و متد Listener آنرا به صورت زیر تکمیل می‌کنم:

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

        if (Build.VERSION.SDK_INT >= 26) {

            mVibrator.vibrate(VibrationEffect.createWaveform(vibratePattern, -1));

        } else {

            mVibrator.vibrate(vibratePattern, -1);

        }

    }
});
```

VibrationEffect.createWaveform یک لرزش الگو دار ایجاد می‌کند. واژه Wave به معنی موج است. یعنی لرزش را به صورت موجی تعریف می‌کند. ورودی نخست آن از جنس long[] است که قبلا تعریف کردیم. ورودی دوم تکرار یا عدم تکرار لرزش را تعیین می‌کند. چنانچه ۱- وارد شود لرزش تکرار نخواهد شد اما در صورت تعریف ۱، الگو تا زمانی که دستور لغو ارسال نشود تکرار خواهد می‌شود.

در اینجا هم مانند دکمه قبل، برای دیوایس‌های پایینتر از Oreo متد به صورت عادی تعریف شده که ورودی‌های آن در هر دو حالت یکسان است.

البته در اندروید ۱۰ (Android Q) تعدادی الگو یا افکت پیش فرض برای ویبره تعریف شده که به صورت زیر قابل استفاده است:



توسط VibrationEffect.createPredefined از ۴ افکت از پیش تعیین شده شامل EFFECT_CLICK، EFFECT_DOUBLE_CLICK، EFFECT_HEAVY_CLICK و EFFECT_TICK می‌توانیم استفاده کنیم.

شرط درون Listener دکمه را به اینصورت تکمیل می‌کنم:

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

        if  (Build.VERSION.SDK_INT >= 29) {


mVibrator.vibrate(VibrationEffect.createPredefined(VibrationEffect.EFFECT_DOUBLE_CLICK))
;

        } else if (Build.VERSION.SDK_INT >= 26) {

            mVibrator.vibrate(VibrationEffect.createWaveform(vibratePattern, -1));

        } else {

            mVibrator.vibrate(vibratePattern, -1);

        }

    }
```

با اجرای مجدد پروژه و کلیک روی دکمه دوم، چنانچه دیوایس از اندروید ۱۰ و یا بالاتر برخوردار باشد، قسمت نخست شرط اجرا شده و لرزش از نوع EFFECT_DOUBLE_CLICK خواهد بود. یعنی یک لرزش شبیه به دابل کلیک. در غیر اینصورت دو قسمت دیگر شرط بر اساس ورژن اندروید اجرا خواهند شد. البته غیر از افکت دابل کلیک، سایر موارد تفاوت چندانی با یکدیگر ندارند!

کد کامل اکتیویتی:

activity_main.xml

```
<?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"
    tools:context=".MainActivity">

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="ویبره ساده"
        android:id="@+id/simple_vibrator" />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="ویبره الگو دار"
        android:id="@+id/pattern_vibrator" />

</LinearLayout>
```

MainActivity.java

```
package ir.android_studio.vibrator;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.os.Build;
import android.os.Bundle;
import android.os.VibrationEffect;
import android.os.Vibrator;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    Button simpleButton, patternButton;
    Vibrator mVibrator;
    long[] vibratePattern = {0,200,400,200,300,200,500,200,500,200,200,1000};

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mVibrator = (Vibrator) getSystemService(Context.VIBRATOR_SERVICE);
        simpleButton = findViewById(R.id.simple_vibrator);
        patternButton = findViewById(R.id.pattern_vibrator);

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

                if (Build.VERSION.SDK_INT >= 26) {

                    mVibrator.vibrate(VibrationEffect.createOneShot(300,
VibrationEffect.DEFAULT_AMPLITUDE));

                } else {

                    mVibrator.vibrate(300);

                }

            }
        });

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

                if  (Build.VERSION.SDK_INT >= 29) {


mVibrator.vibrate(VibrationEffect.createPredefined(VibrationEffect.EFFECT_DOUBLE_CLICK))
;

                } else if (Build.VERSION.SDK_INT >= 26) {

                    mVibrator.vibrate(VibrationEffect.createWaveform(vibratePattern, -
1));

                } else {

                    mVibrator.vibrate(vibratePattern, -1);

                }

            }
        });

    }
}
```

لغو ویبره

امکان لغو و غیر فعال کردن ویبره در حال اجرا نیز وجود دارد. یک الگو دیگر به صورت زیر تعریف می‌کنم:

```
long[] vibratePatternTwo = {0,200,0};
```

الگو فوق در ابتدا و انتها تاخیر ۰ میلی ثانیه‌ای دارد بنابراین در صورت تکرار شدن الگو، یک لرزش ممتد و بی نهایت را شاهد خواهیم بود. اینکه عدد دوم چه عددی باشد اهمیتی نداشته و صرفا باید بزرگتر از ۰ باشد.

قسمت اول شرط در دکمه دوم را غیر فعال می‌کنم تا صرفا الگویی که خودم تعریف کرده‌ام اجرا شود:

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

        /* if  (Build.VERSION.SDK_INT >= 29) {


mVibrator.vibrate(VibrationEffect.createPredefined(VibrationEffect.EFFECT_DOUBLE_CLICK))
;

        } else */ if (Build.VERSION.SDK_INT >= 26) {

            mVibrator.vibrate(VibrationEffect.createWaveform(vibratePatternTwo, 0));

        } else {

            mVibrator.vibrate(vibratePatternTwo, 0);

        }

    }
});
```

الگوی دوم یعنی vibratePatternTwo به عنوان ورودی نخست و عدد صفر را به عنوان ورودی دوم جایگزین کردم. بنابراین بر خلاف قسمت قبل، الگو بی نهایت تکرار خواهد شد.

یک دکمه دیگر با نام "لغو" به layout اضافه کرده و به صورت زیر در اکتیویتی تعریف می‌کنم:

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

        if (mVibrator.hasVibrator()) {
            mVibrator.cancel();
        }

    }
});
```

متد ()hasVibrator بررسی می‌کند آیا ویبره‌ای از سمت mVibrator در حال اجراست یا خیر. چنانچه این شرط مقدار true برگرداند یعنی پاسخ مثبت است و بنابراین متد ()cancel اجرا خواهد شد که باعث لغو ویبره می‌شود.

با اجرای مجدد پروژه و کلیک روی دکمه دوم، ویبره ممتد و بی نهایت اجرا خواهد شد که با کلیک روی دکمه سوم، لغو می‌شود.

کد کامل اکتیویتی:

activity_main.xml

```
<?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"
    tools:context=".MainActivity">

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="ویبره ساده"
        android:id="@+id/simple_vibrator" />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="ویبره الگ ودار"
        android:id="@+id/pattern_vibrator" />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="لغو"
        android:id="@+id/cancel_vibrator" />

</LinearLayout>
```

MainActivity.java

```
package ir.android_studio.vibrator;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.os.Build;
import android.os.Bundle;
import android.os.VibrationEffect;
import android.os.Vibrator;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    Button simpleButton, patternButton, cancelButton;
    Vibrator mVibrator;
    long[] vibratePattern = {0,200,400,200,300,200,500,200,500,200,200,1000};
    long[] vibratePatternTwo = {0,200,0};
```

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

        mVibrator = (Vibrator) getSystemService(Context.VIBRATOR_SERVICE);
        simpleButton = findViewById(R.id.simple_vibrator);
        patternButton = findViewById(R.id.pattern_vibrator);
        cancelButton = findViewById(R.id.cancel_vibrator);

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

                if (Build.VERSION.SDK_INT >= 26) {

                    mVibrator.vibrate(VibrationEffect.createOneShot(300,
VibrationEffect.DEFAULT_AMPLITUDE));

                } else {

                    mVibrator.vibrate(300);

                }

            }
        });

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

                /* if  (Build.VERSION.SDK_INT >= 29) {


mVibrator.vibrate(VibrationEffect.createPredefined(VibrationEffect.EFFECT_DOUBLE_CLICK))
;

                } else */ if (Build.VERSION.SDK_INT >= 26) {

                    mVibrator.vibrate(VibrationEffect.createWaveform(vibratePatternTwo,
0));

                } else {

                    mVibrator.vibrate(vibratePatternTwo, 0);

                }

            }
        });

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

```
                if (mVibrator.hasVibrator()) {
                    mVibrator.cancel();
                }

            }
        });

    }
}
```

مطالعه بیشتر:

https://developer.android.com/reference/android/os/Vibrator

https://developer.android.com/reference/android/os/VibrationEffect

توجه: سورس پروژه درون پوشه Exercises قرار دارد