آموزش برنامه نویسی اندروید
در محیط اندروید استودیو

پخش فایل‌های صوتی در اندروید

مدرس : سیدمهدی مطهری

www.android-studio.ir

به نام خدا

Android MediaPlayer

www.android-studio.ir

در این جلسه ابتدا با کلاس MediaPlayer در اندروید آشنا شده سپس در قالب یک پروژه ساده به بررسی متدهای این کلاس برای مدیریت پخش فایل‌های صوتی در اندروید می‌پردازیم.

معرفی MediaPlayer در اندروید

پخش صوت یکی از قابلیت‌هایی است که امروزه کاربرد گسترده‌ای در تلفن‌های هوشمند دارد. از جمله این کاربردها می‌توان به برنامه‌های پخش موسیقی و یا تقویم‌ها اشاره کرد. کاربر توسط یک برنامه مدیریت فایل‌های صوتی می‌تواند آهنگ‌ها و آلبوم‌های موسیقی را از روی کارت حافظه دیوایس اندرویدی خود اجرا و پخش کند. همچنین قابلیت‌های دیگری در این برنامه‌ها وجود دارند از جمله تغییر فایل صوتی در حال پخش، نمایش مدت زمان هر آهنگ یا فایل صوتی، نمایش موقعیت فعلی پخش صوت (اینکه چه مدت زمانی از شروع سپری شده و چه زمانی باقی مانده) و... . همچنین در یک برنامه تقویم کاربر می‌تواند قابلیت پخش اذان را فعال کند. از دیگر کاربردها می‌توان به پخش صداهای مختلف در یک بازی اشاره کرد. قبلا در آموزش ساخت Splash Screen در اندروید هم از MediaPlayer برای پخش یک صوت هنگام اجرای صفحه اسپلش استفاده کرده بودیم.

در اندروید روش‌های متفاوتی برای اضافه کردن فایل‌های صوتی به برنامه و پخش آنها وجود دارد. یکی از این روش‌ها استفاده از MediaPlayer در اندروید است. با استفاده از این کلاس می‌توانیم مدیریت

کاملی بر روی اجرا و پخش فایل‌های صوتی در اپلیکیشن اندرویدی خود داشته باشیم. قابلیت‌هایی مانند پخش (play)، مکث (pause)، توقف (stop)، پخش دسته جمعی فایل‌ها، نمایش مدت زمان فایل صوتی، نمایش موقعیت زمانی فعلی فایل صوتی در حال پخش، نمایش اطلاعات مربوط به فایل و... .

ابتدا عملکرد تعدادی از متدهای MediaPlayer را بررسی می‌کنیم:

۱: start(): با فراخوانی این متد، فایل صوتی از ابتدا شروع به پخش می‌کند.

۲: pause(): با فراخوانی این متد، در صورتی که قبلا پخش یک فایل صوتی توسط متد start() آغاز شده باشد، متوقف می‌شود. چنانچه مجدد متد start() فراخوانی شود فایل صوتی از جایی که قبلا متوقف شده آغاز خواهد شد و نه از ابتدای فایل.

۳: reset(): همانطور که از نام آن پیداست، MediaPlayer را ریست می‌کند.

۴: isPlaying(): مقدار true یا false برمی‌گرداند و نشان می‌دهد که آیا موزیکی در حال پخش هست یا نه. بخ عنوان مثال می‌توانیم توسط این متد بررسی کنیم فقط در صورتی متد pause() فراخوانی شود که isPlaying() مقدار true برگرداند.

۵: seekTo(): این متد یک مقدار int می‌گیرد و فایل صوتی در حال پخش را به آن نقطه می‌برد. به عنوان مثال اگر فایل صوتی در حال پخش در موقعیت ۱ دقیقه و ۱۰ ثانیه قرار دارد و توسط این متد مقدار ۲۶۰۰۰ میلی ثانیه به MediaPlayer ارسال شود، فایل صوتی از ثانیه ۲۶ شروع به پخش می‌کند.

۶: getCurrentPosition (): موقعیت فعلی پخش را بر حسب میلی ثانیه برمی‌گرداند.

۷: getDuration(): این متد مدت زمان فایل صوتی را بر حسب میلی ثانیه برمی‌گرداند.

۸: release(): با استفاده از این متد فایل‌های صوتی که به MediaPlayer پاس داده شده به ترتیب و پشت سر هم پخش می‌شوند.

۹: setVolume(): برای تنظیم کم و زیاد صدا استفاده می‌شود.

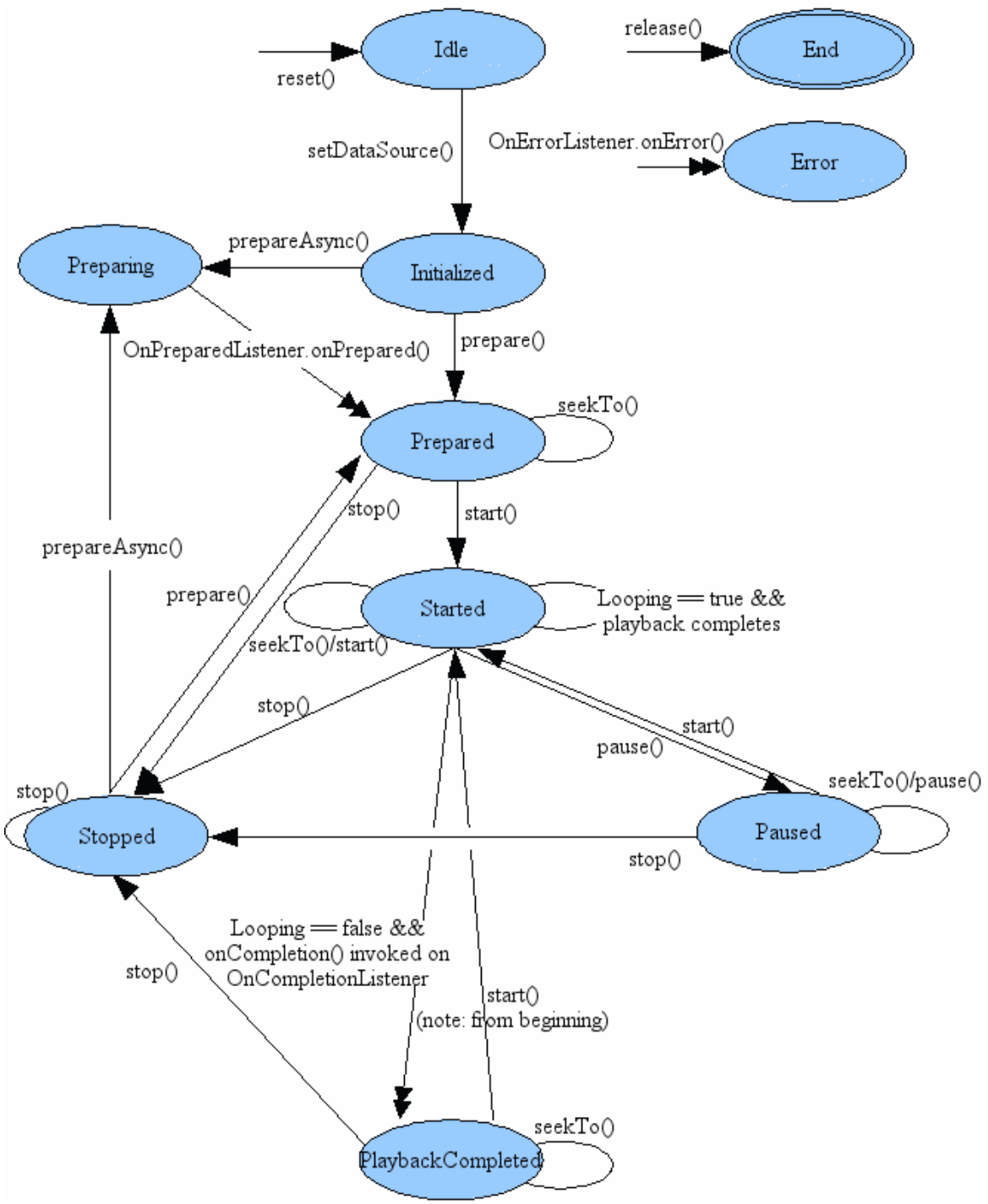
۱۰: selectTrack(): این متد یک مقدار int می‌گیرد که برای انتخاب یک فایل صوتی از یک لیست و پخش آن استفاده می‌شود.

۱۱: selectTrack(): این متد اطلاعات یک تِرَک یا همان فایل صوتی را برمی‌گرداند.

۱۲: setLooping(): لغت Loop به معنی حلقه تکرار است. ورودی این متد از نوع boolean بوده و مقدار true یا false می‌گیرد. چنانچه مقدار true به این متد داده شود فایل صوتی بی نهایت مرتبه تکرار خواهد شد تا زمانی که توسط pause() یا stop() متوقف شود.

نمودار وضعیت MediaPlayer

تصویر زیر یک نمای کلی از وضعیت هرکدام از متدهای MediaPlayer در اندروید را نشان می‌دهد:



اگر نمودار بالا برایتان قابل درک نیست نگران نباشید. ادامه مبحث را دنبال کنید.

پروژه کار با کلاس MediaPlayer اندروید

می‌خواهم در قالب یک تمرین ساده، تعدادی از متدهایی که در ابتدای جلسه معرفی کردم را استفاده کنم تا با کارکرد هرکدام بهتر آشنا شوید.

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

در قدم نخست در Layout اکتیویتی یک TextView و شش Button تعریف کردم که هرکدام از دکمه‌ها یک متد را فراخوانی می‌کند. از TextView هم برای نمایش مدت زمان فایل صوتی، موقعیت فعلی پخش و وضعیت پخش استفاده خواهیم کرد.

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">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="محل نمایش متن"
        android:textAlignment="center"
        android:id="@+id/txt_view" />

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

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

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="توقف"
        android:id="@+id/stop_btn" />
```

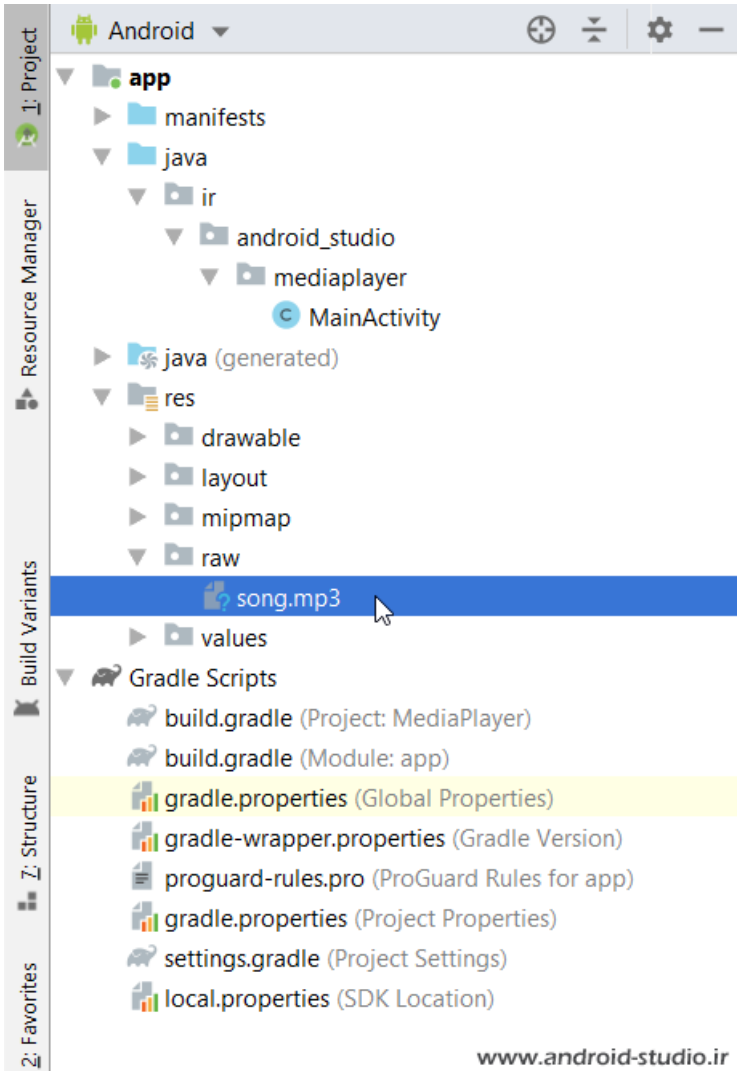
```
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="پرش به ثانیه ۲۰"
        android:id="@+id/seekto_btn" />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="وضعیت پخش"
        android:id="@+id/isplaying_btn" />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="موقعیت فعلی"
        android:id="@+id/position_btn" />

</LinearLayout>
```

قبل از اینکه کدهای اکتیویتی را بنویسم، مطابق آنچه قبلا در آموزش Splash Screen یا آموزش ساخت Service در برنامه نویسی اندروید گفته شد، یک پوشه با نام raw به پوشه‌ی res پروژه اندرویدی اضافه کرده و یک فایل با فرمت mp3. درون آن قرار می‌دهم:



نکته: برای اضافه کردن فایل و یا پوشه‌ای مانند raw الزامی به انجام اینکار در محیط اندروید استودیو نیست بلکه به راحتی و در محیط سیستم عامل می‌توان در محل قرارگیری پروژه، درون پوشه‌ی res یک پوشه جدید با نام raw ایجاد کرد و فایل صوتی را به آن انتقال داد. بعد از برگشت به محیط اندروید استودیو بلافاصله ساختار پروژه بروزرسانی شده و فولدر و فایل‌های اضافه شده مشاهده می‌شود.

حالا نوبت به تکمیل کدهای اکتیویتی می‌رسد:

MainActivity.java

```
package ir.android_studio.mediaplayer;

import androidx.appcompat.app.AppCompatActivity;

import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
import java.util.concurrent.TimeUnit;

public class MainActivity extends AppCompatActivity {

    private TextView txtView;
    private Button startBtn, pauseBtn, stopBtn, seektoBtn, isPlayingBtn, positionBtn;
    private MediaPlayer medPlayer;
    private int timeInt;

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

        txtView = findViewById(R.id.txt_view);
        startBtn = findViewById(R.id.start_btn);
        pauseBtn = findViewById(R.id.pause_btn);
        stopBtn = findViewById(R.id.stop_btn);
        seektoBtn = findViewById(R.id.seekto_btn);
        isPlayingBtn = findViewById(R.id.isplaying_btn);
        positionBtn = findViewById(R.id.position_btn);

        if (medPlayer == null) {
                medPlayer = MediaPlayer.create(this, R.raw.song);
        }

        medPlayer.setLooping(true);

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

                medPlayer.start();

                Toast.makeText(MainActivity.this, "موزیک در حال پخش",
Toast.LENGTH_SHORT).show();

                timeInt = medPlayer.getDuration();
                audioTime();

            }
        });

        pauseBtn.setOnClickListener(new View.OnClickListener() {
```

```
            @Override
            public void onClick(View view) {

                medPlayer.pause();

                Toast.makeText(MainActivity.this, "موزیک موقتا متوقف شد",
Toast.LENGTH_SHORT).show();

            }
        });

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

                medPlayer.stop();

                Toast.makeText(MainActivity.this, "موزیک متوقف شد",
Toast.LENGTH_SHORT).show();

            }
        });

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

                medPlayer.seekTo(20000);

                Toast.makeText(MainActivity.this, "موزیک به ثانیه ۲۰ منتقل شد",
Toast.LENGTH_SHORT).show();

            }
        });

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

                if (medPlayer.isPlaying()) {

                    txtView.setText("موزیک در حال پخش است");

                } else {

                    txtView.setText("موزیک در حال پخش نیست");

                }

            }
        });

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

```
                timeInt = medPlayer.getCurrentPosition();
                audioTime();

            }
        });

    }

    public void audioTime() {

        String time = String.format("%02d min, %02d sec",
                TimeUnit.MILLISECONDS.toMinutes(timeInt),
                TimeUnit.MILLISECONDS.toSeconds(timeInt) -
                TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(timeInt))
        );

        txtView.setText(time);

    }
}
```

ابتدا یک نمونه از کلاس MediaPlayer با نام medPlayer ساختم. داخل متد onCreate توسط MediaPlayer.create فایل صوتی با نام song را به medPlayer معرفی کردم. ورودی نخست کانتکست و ورودی دوم محل قرارگیری فایل صوتی است. این خط را داخل یک شرط قرار دادم تا ابتدا بررسی کند تنها در صورتی MediaPlayer را create کند (یعنی بسازد) که اینکار قبلا انجام نشده و medPlayer برابر null باشد.

در خط بعد برای متد ()setLooping مقدار true قرار دادم تا پخش موزیک تا زمانی که متوقف نشده تکرار شود. برای دکمه شروع ابتدا متد ()start تعریف شده که پخش فایل صوتی را آغاز می‌کند.

یک متغیر از جنس int با نام timeInt در کلاس اکتیویتی تعریف کردم که برای ذخیره مقادیر مربوط به زمان فایل صوتی و موقعیت فعلی پخش از آن استفاده می‌کنم.

```
timeInt = medPlayer.getDuration();
```

در اینجا مدت زمان فایل صوتی توسط getDuration از medPlayer گرفته شده و در متغیر timeInt با واحد میلی ثانیه ذخیره می‌گردد. در خط بعد متدی با نام ()audioTime فراخوانی می‌شود که درون اکتیویتی و بعد از ()onCreate آنرا تعریف کرده‌ام:

```
public void audioTime() {

    String time = String.format("%02d min, %02d sec",
            TimeUnit.MILLISECONDS.toMinutes(timeInt),
            TimeUnit.MILLISECONDS.toSeconds(timeInt) -
            TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(timeInt))
    );

    txtView.setText(time);

}
```

در این تابع ابتدا مقدار ذخیره شده در timeInt توسط String.format() به فرمتی تبدیل شده که در آن مدت زمان موزیک بر حسب دقیقه و ثانیه تفکیک شده است. خروجی این تبدیل در time ذخیره شده و سپس توسط setText به TextView ارسال می‌شود.

بنابراین با کلیک روی دکمه شروع، ابتدا پخش صوت آغاز شده و سپس مدت زمان موزیک در TextView نمایش داده می‌شود.

دکمه‌های مکث و توقف نیاز به توضیحی ندارند. در دکمه پرش برای متد seekTo() مقدار ۲۰۰۰۰ میلی ثانیه (۲۰ ثانیه) تعریف شده. برای دکمه وضعیت پخش یک شرط تعریف شده که بررسی می‌کند چنانچه isPlaying برقرار باشد (یعنی true برگرداند) یک پیغام مناسب با آن و در صورت برقرار نبودن نیز یک پیغام دیگر نمایش دهد. برای دکمه موقعیت فعلی هم ابتدا موقعیت فعلی پخش صوت بر حسب میلی ثانیه از medPlayer دریافت شده و در timeInt ذخیره می‌شود. سپس متد audioTime فراخوانی شده که باعث می‌شود مقدار موجود در timeInt بر حسب دقیقه و ثانیه در TextView نمایش داده شود.

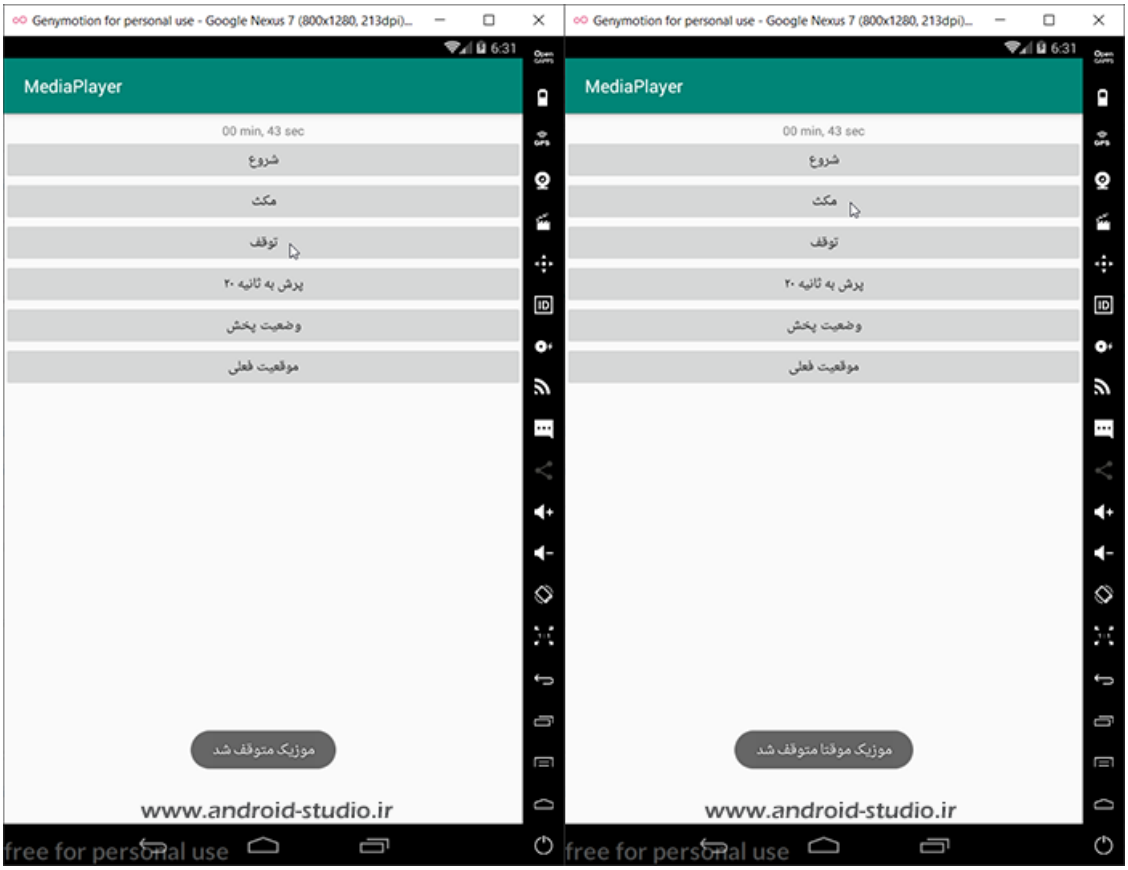
برنامه را اجرا می‌کنم. با کلیک روی دکمه شروع، موزیک پلی شده و همانطور که انتظار می‌رفت مدت زمان موزیک در TextView نمایش داده شد:

با کلیک روی دکمه پرش، موزیک در هر موقعیتی که در حال پخش باشد به ثانیه ۲۰ منتقل خواهد شد:

با کلیک روی دکمه وضعیت پخش، پیغام مناسب در TextView نمایش داده شده است:

دکمه موقعیت فعلی، متد ()getCurrentPosition را فراخوانی می‌کند که در تصویر زیر مشاهده می‌کنید در TextView ثانیه ۴۳ را نمایش می‌دهد.

همچنین دکمه‌های مکث و توقف، عملیات مدنظر را انجام می‌دهند:



MediaPlayer متدهای دیگری برای فراخوانی صوت از URI و URL هم دارد که می‌توان برای پخش مواردی مانند زنگ هشدار (Alarm)، آهنگ زنگ (Ringtone)، پخش آنلاین موزیک و... استفاده کرد اما برای این مبحث به همین حد بسنده می‌کنم. در این جلسه فقط با کلاس MediaPlayer در اندروید و تعدادی از متدهای آن آشنا شدیم. به امید خدا در آینده و در قالب یک آموزش پروژه محور، نحوه ساخت یک Music Player را به طور کامل آموزش خواهم داد.

مطالعه بیشتر:

https://developer.android.com/reference/android/media/MediaPlayer.html

https://developer.android.com/guide/topics/media/mediaplayer

توجه: سورس پروژه درون پوشه Exercises قرار دارد

با ارائه انتقادات و پیشنهادات خود، ما را در ارائه آموزش‌های بهتر یاری فرمائید.
این فایل رایگان بوده و انتشار آن (بدون دخل و تصرف) مانعی ندارد.

www.android-studio.ir