Sunet **RINGTONE** în emulatorul Android

10:01

Start Service
Stop Service

Service has been Started

```kotlin
package com.example.myapplication

import android.util.Log
import android.os.Bundle
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.widget.Button
import android.view.View

// Main Activity class extends the OnClickListener and AppCompactActivity
class MainActivity : View.OnClickListener, AppCompatActivity()  {

    // declaring objects of Button class
    var button1: Button? = null
    lateinit var tag_name : String // tag for logging
    var button2: Button? = null
    lateinit var msg: String

    // overriding the onCreate Function
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        tag_name = "Android Services: "
        button1 = findViewById<View>(R.id.button1) as Button
        msg = "Declaring Listener for button 1"
        button1!!.setOnClickListener(this)
        Log.d(tag_name, msg)

        button2 = findViewById<View>(R.id.button2) as Button
        msg = "Declaring Listener for button 2"
        button2!!.setOnClickListener(this) // declaring listeners for button 2
        Log.d(tag_name, msg)
    }

    override fun onClick(current_view: View) {

        var custom_service = CustomService::class.java
        lateinit var var_intent: Intent

        // check if the current_view is equal to button1
        if (current_view === button1 ) {
            // if yes, then we start the service
            var_intent = Intent(this, custom_service)
            startService(var_intent)
        }
        else if (current_view === button2) {
            // stop the service if current_view is equal to button2
            var_intent = Intent(this, custom_service)
            stopService(var_intent)
        }
    }
}
```

```kotlin
package com.example.myapplication

import android.content.Intent
import android.app.Service
import android.os.IBinder
import android.media.MediaPlayer
import android.provider.Settings
import android.util.Log
import android.widget.Toast

class CustomService : Service() {
   lateinit var ringtone_player:MediaPlayer
    var show_text = ""

    override fun onStartCommand(intent: Intent, flags: Int, Id: Int): Int {
        var ringtone = Settings.System.DEFAULT_RINGTONE_URI // default ringtone of the device
        // creating a MediaPlayer
        ringtone_player = MediaPlayer.create(this, ringtone)

        show_text = "Service has been Started"
        ringtone_player.setLooping(true)

        val duration = Toast.LENGTH_LONG // setting the duration to Long
        ringtone_player.start()     // start the process
        Toast.makeText(this, show_text, duration).show()
        return START_STICKY
    }

    // stop on calling this method
    override fun onDestroy() {
        super.onDestroy()
        // text that needs to be displayed when this function gets invoked
        show_text = "Service has been Stopped"

        ringtone_player.stop() // stop the ringtone_player

        val duration = Toast.LENGTH_LONG // setting the duration to Long

        Toast.makeText(this, show_text, duration).show()

    }
    // overriding the onBind Method
    override fun onBind(intent: Intent): IBinder? {
        //as we don't need to bind anything to this service
        // we are returning null from this method
        return null
    }
}
```

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

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.MyApplication"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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


        <!--Registering the New Service-->
        <service android:name=".CustomService"/>

    </application>
</manifest>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout 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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="48dp"
        android:text="Start Service"
        app:layout_constraintBaseline_toBaselineOf="@+id/button2"
        app:layout_constraintStart_toStartOf="parent" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="29dp"
        android:layout_marginBottom="82dp"
        android:text="Stop Service"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toEndOf="@+id/button1" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

```kotlin
plugins {
    alias(libs.plugins.android.application)
    alias(libs.plugins.kotlin.android)
}

android {
    namespace = "com.example.myapplication"
    compileSdk = 35

    defaultConfig {
        applicationId = "com.example.myapplication"
        minSdk = 33
        targetSdk = 34
        versionCode = 1
        versionName = "1.0"

        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            isMinifyEnabled = false
            proguardFiles(
                getDefaultProguardFile("proguard-android-optimize.txt"),
                "proguard-rules.pro"
            )
        }
    }
    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_1_8
        targetCompatibility = JavaVersion.VERSION_1_8
    }
    kotlinOptions {
        jvmTarget = "1.8"
    }
}

dependencies {

    implementation(libs.androidx.core.ktx)
    /*
    1.  Dependency 'androidx.core:core-ktx:1.15.0' requires libraries and applications that
      depend on it to compile against version 35 or later of the
      Android APIs.

      :app is currently compiled against android-34.
      ..
      Update minCompileSdk in modules with dependencies that require a higher minCompileSdk.
     */

    implementation(libs.androidx.appcompat)
    implementation(libs.material)
    implementation(libs.androidx.activity)
    implementation(libs.androidx.constraintlayout)
    testImplementation(libs.junit)
    androidTestImplementation(libs.androidx.junit)
    androidTestImplementation(libs.androidx.espresso.core)
}
```

Sound Service

http://cti.ubm.ro/cmo/07/example.mp3

```
package com.example.soundservice

import android.content.Intent
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import android.widget.Button

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val playSoundButton: Button = findViewById(R.id.playSoundButton)

        playSoundButton.setOnClickListener {
            // Pornește serviciul care redă sunetul
            val intent = Intent(this, SoundService::class.java)
            startService(intent)
        }
    }
}
```

```
package com.example.soundservice

import android.app.Service
import android.content.Intent
import android.media.MediaPlayer
import android.os.Handler
import android.os.IBinder

class SoundService : Service() {

    private val handler = Handler()
    private lateinit var mediaPlayer: MediaPlayer

    override fun onStartCommand(intent: Intent?, flags: Int, startId: Int): Int {

        // Amână redarea sunetului cu 3 secunde
        handler.postDelayed({
            mediaPlayer = MediaPlayer.create(this, R.raw.example)
           // Adaugă un fișier .mp3 în res/raw

            mediaPlayer.start()

            // Oprește sunetul după ce s-a terminat
            mediaPlayer.setOnCompletionListener {
                stopSelf()
            }
        }, 3000)

        return START_STICKY
    }

    override fun onDestroy() {
        super.onDestroy()
        if (this::mediaPlayer.isInitialized) {
            mediaPlayer.release()
        }
    }

    override fun onBind(intent: Intent?): IBinder? {
        return null
    }
}
```

Foreground SoundService

Serviciile obișnuite (Service) rulează în fundal atâta timp cât nu sunt întrerupte de sistem sau nu sunt oprite explicit (folosind stopSelf() sau stopService()).

Legătura dintre serviciu și aplicație: chiar dacă serviciul este proiectat să ruleze în fundal, el poate fi întrerupt dacă sistemul consideră că resursele sunt limitate sau dacă aplicația care a pornit serviciul este închisă.

Android recent folosește restricții pe fundal: Începând cu Android 8 (API 26), serviciile normale sunt supuse unor restricții stricte pentru a conserva bateria și performanța:
Dacă o aplicație care a pornit un serviciu este închisă (terminată complet), serviciile obișnuite sunt și ele terminate, deoarece aplicația nu mai este activă.

Pentru rularea continuă a serviciului chiar dacă aplicația este închisă, se poate transforma serviciul anterior într-un Foreground Service. Un Foreground Service este un tip de serviciu care rulează în prim-plan și afișează o notificare permanentă pentru utilizator.

Pentru a folosi un Foreground Service în Android, trebuie să declari permisiunea în fișierul AndroidManifest.xml.

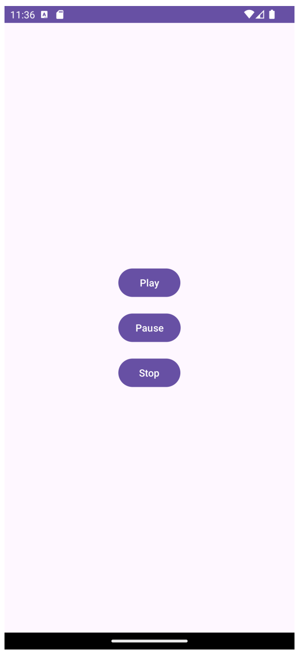
```
    <uses-permission android:name="android.permission.FOREGROUND_SERVICE" />
```

```
package com.example.foregroundsoundservice

import android.content.Intent
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import android.widget.Button

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val playSoundButton: Button = findViewById(R.id.playSoundButton)

        playSoundButton.setOnClickListener {
            // Pornește serviciul care redă sunetul
            val intent = Intent(this, ForeSoundService::class.java)
            startService(intent)
        }
    }
}
```

```
package com.example.foregroundsoundservice


import android.app.Notification
import android.app.NotificationChannel
import android.app.NotificationManager
import android.app.Service
import android.content.Intent
import android.media.MediaPlayer
import android.os.Build
import android.os.Handler
import android.os.IBinder
import androidx.core.app.NotificationCompat

class ForeSoundService : Service() {

    private val handler = Handler()
    private lateinit var mediaPlayer: MediaPlayer

    override fun onStartCommand(intent: Intent?, flags: Int, startId: Int): Int {
        // Creează un canal de notificare pentru Android 8+ (obligatoriu pentru foreground services)
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            val channel = NotificationChannel(
                "SoundServiceChannel",
                "Sound Service Channel",
                NotificationManager.IMPORTANCE_LOW
            )
            val manager = getSystemService(NotificationManager::class.java)
            manager?.createNotificationChannel(channel)
        }

        // Creează notificarea pentru Foreground Service
        val notification: Notification = NotificationCompat.Builder(this, "SoundServiceChannel")
            .setContentTitle("Sound Service")
            .setContentText("Playing sound...")
            .setSmallIcon(android.R.drawable.ic_media_play)
            .build()

        startForeground(1, notification)

        // Redă sunetul după 3 secunde
        handler.postDelayed({
            mediaPlayer = MediaPlayer.create(this, R.raw.example) // Adaugă fișier audio
            mediaPlayer.start()

            // Oprește serviciul după terminarea sunetului
            mediaPlayer.setOnCompletionListener {
                stopForeground(true) // Elimină notificarea
                stopSelf() // Oprește serviciul
            }
        }, 3000)

        return START_STICKY
    }

    override fun onDestroy() {
        super.onDestroy()
        if (this::mediaPlayer.isInitialized) {
            mediaPlayer.release()
        }
    }

    override fun onBind(intent: Intent?): IBinder? {
        return null
    }
}
```

Pentru a opri un Foreground Service la apăsarea unui buton, trebuie să adaugi un mecanism care trimite un semnal către serviciu, indicând că acesta ar trebui să se oprească.

http://cti.ubm.ro/cmo/07/StopSoundService.zip


11:36
Play
Pause
Stop