Android Service

XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/a
ndroid"
    android:orientation="vertical"
android:layout_width="match_parent"
    android:layout_height="match_parent">
    <Button
        android:id="@+id/btnStart"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="start"
        android:layout_marginLeft="130dp"
        android:layout_marginTop="150dp"
        android:text="Start Service"/>
    <Button
        android:id="@+id/btnstop"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="stop"
        android:layout_marginLeft="130dp"
        android:layout_marginTop="20dp"
        android:text="Stop Service"/>
</LinearLayout>
```

Kotlin

MainActivity.kt

```
package com.example.androidserviceeg
```

```
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.content.Intent
import android.view.View



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

    // Start the service
    fun start(view: View?) {

        startService(Intent(this,
MyService::class.java))
    }

    // Stop the service
    fun stop(view: View?) {
        stopService(Intent(this,
MyService::class.java))
    }

}
```

MyService.kt

```
package com.example.androidserviceeg

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

class MyService : Service() {
    private lateinit var player: MediaPlayer


override fun onBind(intent: Intent): IBinder {
  TODO("Return the communication channel to the service.")
}


    override fun onStartCommand(intent: Intent?,
flags: Int, startId: Int): Int {
        player = MediaPlayer.create(this,
Settings.System.DEFAULT_RINGTONE_URI)
//  player = MediaPlayer.create(this, R.raw.test)


        // This will play the ringtone continuously
until we stop the service.
        player.isLooping = true
        // It will start the player
        player.start()
        Toast.makeText(this, "Service Started",
Toast.LENGTH_LONG).show()
        return START_STICKY
    }

    override fun onDestroy() {
        super.onDestroy()
        // Stopping the player when service is
destroyed
        player.stop()
        Toast.makeText(this, "Service Stopped",
Toast.LENGTH_LONG).show()
    }
}
```