To learn about services

<https://developer.android.com/guide/components/services>

* Go to your music folder and copy any song
* Create a raw folder
* Right click on res folder, New->Android Resource Directory, set the resource type to raw
* Paste the song (rename the song with just lower case letters)

*<?***xml version="1.0" encoding="utf-8"***?>*

<**RelativeLayout 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"**

**tools:context=".MainActivity"**>

<**Button**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:text="start"**

**android:onClick="start"**

**android:layout\_marginTop="100dp"**/>

<**Button**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:text="stop"**

**android:onClick="stop"**

**android:layout\_marginTop="200dp"**/>

</**RelativeLayout**>

**package** com.example.archana.a27\_services;

**import** android.content.Intent;

**import** android.support.v7.app.AppCompatActivity;

**import** android.os.Bundle;

**import** android.view.View;

**import** android.widget.TextView;

**public class** MainActivity **extends**

AppCompatActivity {

@Override

**protected void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.***activity\_main***);

}

**public void** start(View v)

{

Intent i=**new** Intent(**this**,

MyService.**class**);

startService(i);

}

**public void** stop(View v)

{

Intent i=**new** Intent(**this**,

MyService.**class**);

stopService(i);

}

}

Create a Service class as, under java, right click package on the package, New-> Service->Service

**package** com.example.archana.a27\_services;

**import** android.app.Service;

**import** android.content.Intent;

**import** android.media.MediaPlayer;

**import** android.os.IBinder;

**import** android.widget.Toast;

**public class** MyService **extends** Service {

MediaPlayer **myPlayer**;

@Override

**public** IBinder onBind(Intent intent) {

Toast.*makeText*(**this**,**"onBind"**,

Toast.***LENGTH\_SHORT***).show();

*//* ***TODO: Return the communication channel to the service.***

**throw new** UnsupportedOperationException

(**"Not yet implemented"**);

}

@Override

**public void** onCreate() {

**super**.onCreate();

Toast.*makeText*(**this**,**"onCreate"**,

Toast.***LENGTH\_SHORT***).show();

**myPlayer** = MediaPlayer.*create*

(**this**, R.raw.***ajabsi*** );

*// myPlayer.setLooping(false); // Set looping*

}

@Override

**public int** onStartCommand(Intent intent, **int** flags,

**int** startId) {

Toast.*makeText*(**this**,**"onStartCommand"**,

Toast.***LENGTH\_SHORT***).show();

**myPlayer**.start();

**return *START\_STICKY***;

*/\**

*START\_STICKY : Using this return value,*

*if the OS kills our Service it will recreate it*

*but the Intent that was sent*

*to the Service isn’t redelivered.*

*In this way the Service is always running*

*START\_NOT\_STICKY:*

*If the OS kills the Service it won’t recreate*

*it until the client calls*

*explicitly onStart command*

*START\_REDELIVER\_INTENT: It is similar to*

*the START\_STICKY and in this case,*

*the Intent will be*

*redelivered to the service.*

*\*/*

}

@Override

**public void** onDestroy() {

**super**.onDestroy();

Toast.*makeText*(**this**,**"OnDestroy"**,

Toast.***LENGTH\_SHORT***).show();

**myPlayer**.stop();

}

}