1. Create new resource directory **‘raw’** under **‘res’** folder and copy the music file into that folder.

*activity\_main.xml*

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

<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"**

**xmlns:tools="http://schemas.android.com/tools"**

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

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

**android:orientation="vertical"** >

<**TextView**

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

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

**android:gravity="center\_horizontal"**

**android:text="MusicPlayer With BoundService"** />

<**LinearLayout**

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

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

**android:orientation="vertical"**

**android:layout\_marginTop="50dp"**

**android:layout\_marginLeft="100dp"**>

<**Button**

**android:id="@+id/startServiceButton"**

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

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

**android:text="Open"** />

<**Button**

**android:id="@+id/fastBackward"**

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

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

**android:layout\_marginTop="10dp"**

**android:text="Rewind"** />

<**Button**

**android:id="@+id/fastForward"**

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

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

**android:layout\_marginTop="10dp"**

**android:text="Fast forward"** />

<**Button**

**android:id="@+id/stopServiceButton"**

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

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

**android:layout\_marginTop="10dp"**

**android:text="Close"** />

</**LinearLayout**>

</**LinearLayout**>

MainAcitvity.java

**package** com.example.jevitha.a24\_2\_service\_musicplayer\_bound;

**import** android.content.ComponentName;

**import** android.content.Context;

**import** android.content.Intent;

**import** android.content.ServiceConnection;

**import** android.os.IBinder;

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

**import** android.os.Bundle;

**import** android.view.View;

**import** android.widget.Button;

**public class** MainActivity **extends** AppCompatActivity **implements** View.OnClickListener {

**private** Button startServiceButton, stopServiceButton, fastBackward,

fastForward;

**private** Intent intentService;

**private** SimpleAudioService simpleAudioService;

**boolean** serviceStarted = **false**;

@Override

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

**super**.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

init();

}

*/\*\**

*\* Instantiation of components*

*\*/*

**private void** init() {

startServiceButton = (Button) findViewById(R.id.startServiceButton);

stopServiceButton = (Button) findViewById(R.id.stopServiceButton);

fastBackward = (Button) findViewById(R.id.fastBackward);

fastForward = (Button) findViewById(R.id.fastForward);

startServiceButton.setOnClickListener(**this**);

stopServiceButton.setOnClickListener(**this**);

fastBackward.setOnClickListener(**this**);

fastForward.setOnClickListener(**this**);

intentService = **new** Intent(**this**, SimpleAudioService.**class**);

}

@Override

**public void** onClick(View v) {

**switch** (v.getId()) {

*/\*\*\**

*\* Open*

*\*/*

**case** R.id.startServiceButton:

startService(intentService);

*/\*\**

*\* Call bindService() to start a bound Service and to bind to the Service,*

*\**

*\* intentService - Intent to Service class*

*\**

*\* serviceConnection - Interface for monitoring the state of an application service.*

*\**

*\* The third parameter is a flag indicating options for the binding.*

*\* It should usually be BIND\_AUTO\_CREATE in order to create the service if it's not already alive.*

*\* BIND\_DEBUG\_UNBIND - include debugging help for mismatched calls to unbind.*

*\* BIND\_NOT\_FOREGROUND - don't allow this binding to raise the target service's process to the foreground scheduling priority.*

*\* or 0 for none.*

*\*/*

bindService(intentService, serviceConnection, Context.BIND\_AUTO\_CREATE);

serviceStarted = **true**;

**break**;

*/\*\**

*\* Close*

*\*/*

**case** R.id.stopServiceButton:

**if** (serviceStarted) {

*/\**

*\* When a component has finished using the Service, they call unbindService() to unbind from it.*

*\* When all components have unbound from the Service, it is destroyed.*

*\*/*

unbindService(serviceConnection);

stopService(intentService);

serviceStarted = **false**;

}

**break**;

*/\*\**

*\* Rewind*

*\*/*

**case** R.id.fastBackward:

**if** (serviceStarted) {

simpleAudioService.rewind();

}

**break**;

*/\*\**

*\* Fast forward*

*\*/*

**case** R.id.***fastForward***:

**if** (**serviceStarted**) {

**simpleAudioService**.fastforward();

}

**break**;

**default**:

**break**;

}

}

*/\*\**

*\* ServiceConnection is an object of type ServiceConnection,*

*\* which is an interface for service binding*

*\*/*

**private** ServiceConnection **serviceConnection** = **new** ServiceConnection() {

*/\*\**

*\* Click the open button, will call the serviceConnection object's onServiceConnected method.*

*\* Pass a IBinder object to the method*

*\* The IBinder object is of type SimpleAudioServiceBinder - created in service class.*

*\*/*

@Override

**public void** onServiceConnected(ComponentName name,

IBinder sasBinder) {

**simpleAudioService** = ((SimpleAudioService.

SimpleAudioServiceBinder) sasBinder).getService();

}

*/\*\**

*\* The method is used for service disconnection .*

*\*/*

@Override

**public void** onServiceDisconnected(ComponentName name) {

**simpleAudioService** = **null**;

}

};

}

SimpleAudioService.java

**package** com.example.jevitha.a24\_2\_service\_musicplayer\_bound;

**import** android.app.Service;

**import** android.content.Intent;

**import** android.media.MediaPlayer;

**import** android.os.Binder;

**import** android.os.IBinder;

**import** android.util.Log;

**import** android.widget.Toast;

**public class** SimpleAudioService **extends** Service {

**private** String **TAG** = **"MusicPlayer"**;

**private** MediaPlayer **player**;

*/\**

*\* IBinder defines the interface that components use to interact with the Service.*

*\* Get the interface IBinder by extending the Binder class.*

*\* The client then uses this interface to access the public methods in the Service.*

*\*/*

**public class** SimpleAudioServiceBinder **extends** Binder {

SimpleAudioService getService() {

**return** SimpleAudioService.**this**;

}

}

**private final** IBinder **sasBinder** = **new** SimpleAudioServiceBinder();

*/\**

*\* Implement onBind() which returns an IBinder.*

*\* These are the clients that can bind to a Service:Activities , Services , Content Providers*

*\* You can’t bind to a Service from: Broadcast Receivers*

*\*/*

@Override

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

Toast.*makeText*(**this**, **"Service onBind()"**, Toast.***LENGTH\_SHORT***).show();

**return sasBinder**;

}

@Override

**public boolean** onUnbind(Intent intent) {

Toast.*makeText*(**this**, **"Service onUnBind()"**, Toast.***LENGTH\_SHORT***).show();

**return super**.onUnbind(intent);

}

@Override

**public void** onCreate() {

**super**.onCreate();

Toast.*makeText*(**this**, **"Service onCreate()"**, Toast.***LENGTH\_SHORT***).show();

**player** = MediaPlayer.*create*(**this**, R.raw.***yen***);

}

@Override

**public int** onStartCommand(Intent intent, **int** flags, **int** startId) {

Toast.*makeText*(**this**, **"Service onStartCommand()"**, Toast.***LENGTH\_SHORT***).show();

**if** (!**player**.isPlaying()) {

**player**.start();

}

**return super**.onStartCommand(intent, flags, startId);

}

*/\*\**

*\* Call stopService to stop the service, will call onDestroy () method.*

*\*/*

@Override

**public void** onDestroy() {

**super**.onDestroy();

Toast.*makeText*(**this**, **"Service onDestroy()"**, Toast.***LENGTH\_SHORT***).show();

**if** (**player**.isPlaying()) {

**player**.stop();

}

**player**.release();

}

*/\*\**

*\* Service method: rewind*

*\*/*

**public void** rewind() {

**if** (**player**.isPlaying()) {

**player**.seekTo(**player**.getCurrentPosition() - 2500);

Toast.*makeText*(**this**, **"Rewind to "**+ **player**.getCurrentPosition() , Toast.***LENGTH\_SHORT***).show();

}

}

*/\*\**

*\* Service: fast forward method*

*\*/*

**public void** fastforward() {

**if** (**player**.isPlaying()) {

**player**.seekTo(**player**.getCurrentPosition() + 2500);

Toast.*makeText*(**this**, **"Forward to "**+ **player**.getCurrentPosition() , Toast.***LENGTH\_SHORT***).show();

}

}

}

AndroidManifest.xml

(Will automatically be updated if service is created using app → New → Service → Service)

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

<**manifest xmlns:android="http://schemas.android.com/apk/res/android"**

**package="com.example.jevitha.a24\_2\_service\_musicplayer\_bound"**>

<**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**>

<**service**

**android:name=".SimpleAudioService"**

**android:enabled="true"**

**android:exported="true"**></**service**>

</**application**>

</**manifest**>