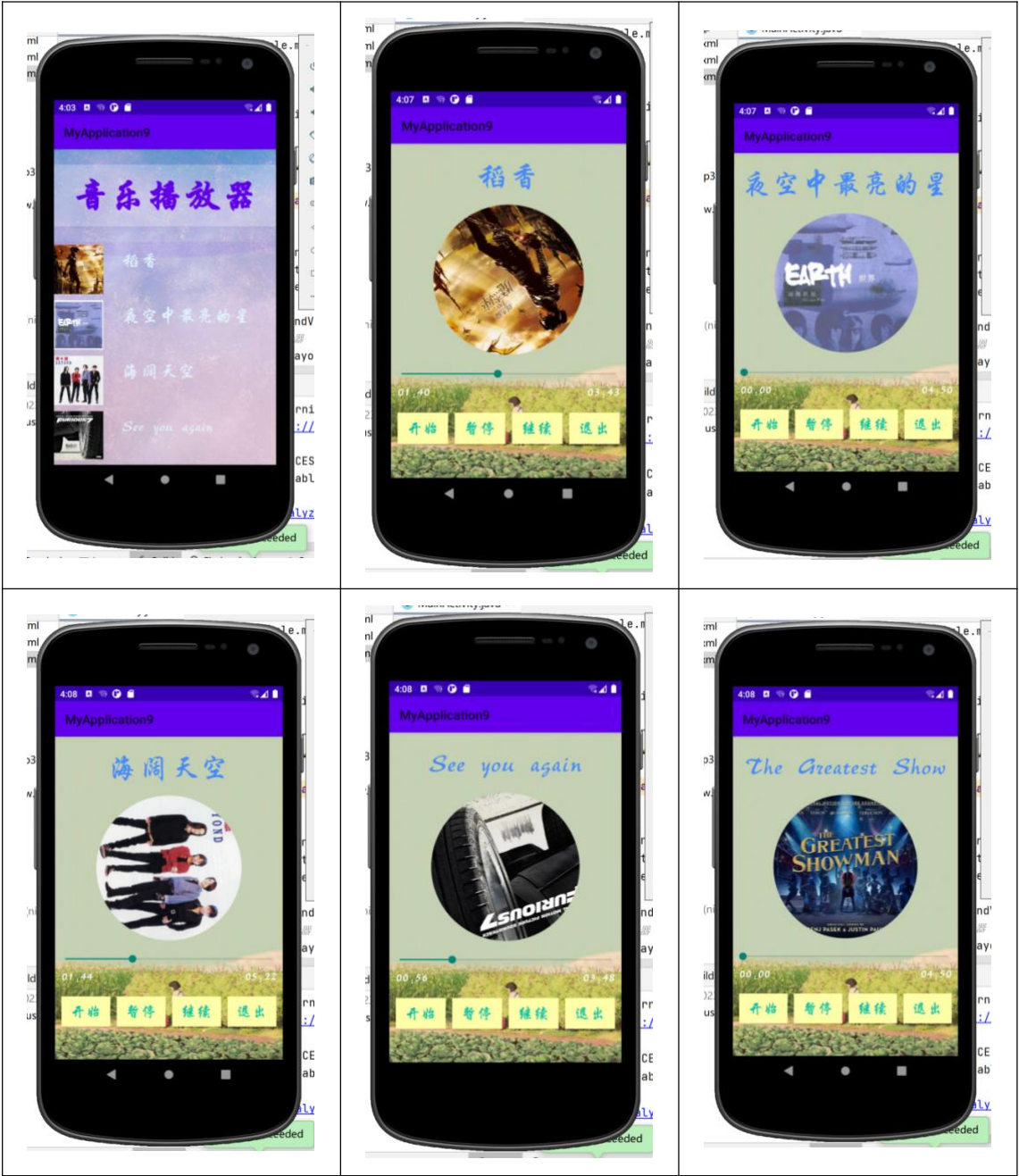


音乐播放器 app 报告

12210159 应逸雯

注：由于文档内容长，增加了目录导航，可根据目录直接跳转查看

界面图片：



代码部分:

界面代码:

in activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:background="@drawable/bg2">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="100dp"
        android:text="音乐播放器"
        android:textSize="60dp"
        android:textColor="#6200EA"
        android:textStyle="bold"
        android:gravity="center"
        android:fontFamily="@font/hanyi"
        android:layout_marginTop="25dp"
        android:layout_marginBottom="25dp"
        android:background="#40FFFFFF"/>

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recycle_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

</LinearLayout>
```

in activity_song1.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"
    tools:context=".song1"
    android:background="@drawable/bg"
    android:orientation="vertical">

    <TextView
```

```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="稻香"
android:gravity="center"
android:layout_marginTop="25dp"
android:textColor="#448AFF"
android:textSize="48sp"
android:fontFamily="@font/hanyi"/>
```

<ImageView

```
android:id="@+id/iv_cover1"
android:layout_width="230dp"
android:layout_height="230dp"
android:layout_gravity="center_horizontal"
android:layout_marginTop="20dp"
android:src="@drawable/img_1" />
```

<SeekBar

```
android:id="@+id/sb1"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="20dp"/>
```

<RelativeLayout

```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_margin="10dp">
```

<TextView

```
android:id="@+id/tv_progress1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="00:00"
android:textColor="@color/white"
android:textSize="20sp"
android:fontFamily="@font/hanyi"
android:textStyle="bold"/>
```

<TextView

```
android:id="@+id/tv_total1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="04:50"
android:layout_alignParentRight="true"
android:textSize="20sp"
```

```
        android:textColor="@color/white"
        android:fontFamily="@font/hanyi"
        android:textStyle="bold"/>
```

```
</RelativeLayout>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_margin="5dp">
```

```
<Button
```

```
    android:id="@+id/btn_play1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="开始"
    android:textSize="25sp"
    android:layout_weight="1"
    android:layout_margin="5dp"
    android:textColor="@color/buttonword"
    android:background="@color/buttonbackground"
    android:fontFamily="@font/hanyi"/>
```

```
<Button
```

```
    android:id="@+id/btn_pause1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="暂停"
    android:textSize="25sp"
    android:layout_weight="1"
    android:layout_margin="5dp"
    android:textColor="@color/buttonword"
    android:background="@color/buttonbackground"
    android:fontFamily="@font/hanyi"/>
```

```
<Button
```

```
    android:id="@+id/btn_continue1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="继续"
    android:textSize="25sp"
    android:layout_weight="1"
    android:layout_margin="5dp"
    android:textColor="@color/buttonword"
    android:background="@color/buttonbackground"
    android:fontFamily="@font/hanyi"/>
```

```
<Button
```

```
        android:id="@+id/btn_exit1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="退出"
        android:textSize="25sp"
        android:layout_weight="1"
        android:layout_margin="5dp"
        android:textColor="@color/buttonword"
        android:background="@color/buttonbackground"
        android:fontFamily="@font/hanyi"/>
```

```
</LinearLayout>
```

```
</LinearLayout>
```

in activity_song2.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"
```

```
    tools:context=".song2"
```

```
    android:background="@drawable/bg"
```

```
    android:orientation="vertical">
```

```
    <TextView
```

```
        android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
```

```
        android:text="夜空中最亮的星"
```

```
        android:gravity="center"
```

```
        android:layout_marginTop="25dp"
```

```
        android:textColor="#448AFF"
```

```
        android:textSize="48sp"
```

```
        android:fontFamily="@font/hanyi"/>
```

```
    <ImageView
```

```
        android:id="@+id/iv_cover2"
```

```
        android:layout_width="230dp"
```

```
        android:layout_height="230dp"
```

```
        android:layout_gravity="center_horizontal"
```

```
        android:layout_marginTop="20dp"
```

```
        android:src="@drawable/img_2" />
```

```
    <SeekBar
```

```
        android:id="@+id/sb2"
```

```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="20dp"/>
```

<RelativeLayout

```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_margin="10dp">
```

<TextView

```
android:id="@+id/tv_progress2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="00:00"
android:textColor="@color/white"
android:textSize="20sp"
android:fontFamily="@font/hanyi"
android:textStyle="bold"/>
```

<TextView

```
android:id="@+id/tv_total2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="04:50"
android:layout_alignParentRight="true"
android:textSize="20sp"
android:textColor="@color/white"
android:fontFamily="@font/hanyi"
android:textStyle="bold"/>
```

</RelativeLayout>

<LinearLayout

```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="horizontal"
android:layout_margin="5dp">
```

<Button

```
android:id="@+id/btn_play2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="开始"
android:textSize="25sp"
android:layout_weight="1"
android:layout_margin="5dp"
android:textColor="@color/buttonword"
```

```
        android:background="@color/buttonbackground"
        android:fontFamily="@font/hanyi"/>
```

```
<Button
```

```
    android:id="@+id/btn_pause2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="暫停"
    android:textSize="25sp"
    android:layout_weight="1"
    android:layout_margin="5dp"
    android:textColor="@color/buttonword"
    android:background="@color/buttonbackground"
    android:fontFamily="@font/hanyi"/>
```

```
<Button
```

```
    android:id="@+id/btn_continue2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="繼續"
    android:textSize="25sp"
    android:layout_weight="1"
    android:layout_margin="5dp"
    android:textColor="@color/buttonword"
    android:background="@color/buttonbackground"
    android:fontFamily="@font/hanyi"/>
```

```
<Button
```

```
    android:id="@+id/btn_exit2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="退出"
    android:textSize="25sp"
    android:layout_weight="1"
    android:layout_margin="5dp"
    android:textColor="@color/buttonword"
    android:background="@color/buttonbackground"
    android:fontFamily="@font/hanyi"/>
```

```
</LinearLayout>
```

```
</LinearLayout>
```

in activity_song3.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"
    tools:context=".song3"
    android:background="@drawable/bg"
    android:orientation="vertical">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="海阔天空"
        android:gravity="center"
        android:layout_marginTop="25dp"
        android:textColor="#448AFF"
        android:textSize="48sp"
        android:fontFamily="@font/hanyi"/>

    <ImageView
        android:id="@+id/iv_cover3"
        android:layout_width="230dp"
        android:layout_height="230dp"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="20dp"
        android:src="@drawable/img_3" />

    <SeekBar
        android:id="@+id/sb3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"/>

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp">

        <TextView
            android:id="@+id/tv_progress3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="00:00"
            android:textColor="@color/white"
            android:textSize="20sp"
            android:fontFamily="@font/hanyi"
            android:textStyle="bold"/>
        <TextView

```



```
        android:id="@+id/tv_total3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="04:50"
        android:layout_alignParentRight="true"
        android:textSize="20sp"
        android:textColor="@color/white"
        android:fontFamily="@font/hanyi"
        android:textStyle="bold"/>
```

```
</RelativeLayout>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_margin="5dp">
```

```
<Button
```

```
    android:id="@+id/btn_play3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="开始"
    android:textSize="25sp"
    android:layout_weight="1"
    android:layout_margin="5dp"
    android:textColor="@color/buttonword"
    android:background="@color/buttonbackground"
    android:fontFamily="@font/hanyi"/>
```

```
<Button
```

```
    android:id="@+id/btn_pause3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="暂停"
    android:textSize="25sp"
    android:layout_weight="1"
    android:layout_margin="5dp"
    android:textColor="@color/buttonword"
    android:background="@color/buttonbackground"
    android:fontFamily="@font/hanyi"/>
```

```
<Button
```

```
    android:id="@+id/btn_continue3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="继续"
    android:textSize="25sp"
```

```

        android:layout_weight="1"
        android:layout_margin="5dp"
        android:textColor="@color/buttonword"
        android:background="@color/buttonbackground"
        android:fontFamily="@font/hanyi"/>
    <Button
        android:id="@+id/btn_exit3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="退出"
        android:textSize="25sp"
        android:layout_weight="1"
        android:layout_margin="5dp"
        android:textColor="@color/buttonword"
        android:background="@color/buttonbackground"
        android:fontFamily="@font/hanyi"/>

```

```

</LinearLayout>

```

```

</LinearLayout>

```

in activity_song4.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"
    tools:context=".song4"
    android:background="@drawable/bg"
    android:orientation="vertical">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="See you again"
        android:gravity="center"
        android:layout_marginTop="25dp"
        android:textColor="#448AFF"
        android:textSize="48sp"
        android:fontFamily="@font/hanyi"/>

    <ImageView
        android:id="@+id/iv_cover4"
        android:layout_width="230dp"
        android:layout_height="230dp"

```

```
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="20dp"
    android:src="@drawable/img_4" />
```

<SeekBar

```
    android:id="@+id/sb4"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"/>
```

<RelativeLayout

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp">
```

<TextView

```
    android:id="@+id/tv_progress4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="00:00"
    android:textColor="@color/white"
    android:textSize="20sp"
    android:fontFamily="@font/hanyi"
    android:textStyle="bold"/>
```

<TextView

```
    android:id="@+id/tv_total4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="04:50"
    android:layout_alignParentRight="true"
    android:textSize="20sp"
    android:textColor="@color/white"
    android:fontFamily="@font/hanyi"
    android:textStyle="bold"/>
```

</RelativeLayout>

<LinearLayout

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_margin="5dp">
```

<Button

```
    android:id="@+id/btn_play4"
    android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
android:text="开始"
android:textSize="25sp"
android:layout_weight="1"
android:layout_margin="5dp"
android:textColor="@color/buttonword"
android:background="@color/buttonbackground"
android:fontFamily="@font/hanyi"/>
```

<Button

```
android:id="@+id/btn_pause4"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="暂停"
android:textSize="25sp"
android:layout_weight="1"
android:layout_margin="5dp"
android:textColor="@color/buttonword"
android:background="@color/buttonbackground"
android:fontFamily="@font/hanyi"/>
```

<Button

```
android:id="@+id/btn_continue4"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="继续"
android:textSize="25sp"
android:layout_weight="1"
android:layout_margin="5dp"
android:textColor="@color/buttonword"
android:background="@color/buttonbackground"
android:fontFamily="@font/hanyi"/>
```

<Button

```
android:id="@+id/btn_exit4"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="退出"
android:textSize="25sp"
android:layout_weight="1"
android:layout_margin="5dp"
android:textColor="@color/buttonword"
android:background="@color/buttonbackground"
android:fontFamily="@font/hanyi"/>
```

</LinearLayout>

</LinearLayout>

in activity_song5.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"
```

```
    tools:context=".song5"
```

```
    android:background="@drawable/bg"
```

```
    android:orientation="vertical">
```

```
    <TextView
```

```
        android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
```

```
        android:text="The Greatest Show"
```

```
        android:gravity="center"
```

```
        android:layout_marginTop="25dp"
```

```
        android:textColor="#448AFF"
```

```
        android:textSize="48sp"
```

```
        android:fontFamily="@font/hanyi"/>
```

```
    <ImageView
```

```
        android:id="@+id/iv_cover5"
```

```
        android:layout_width="230dp"
```

```
        android:layout_height="230dp"
```

```
        android:layout_gravity="center_horizontal"
```

```
        android:layout_marginTop="20dp"
```

```
        android:src="@drawable/img_5" />
```

```
    <SeekBar
```

```
        android:id="@+id/sb5"
```

```
        android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
```

```
        android:layout_marginTop="20dp"/>
```

```
    <RelativeLayout
```

```
        android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
```

```
        android:layout_margin="10dp">
```

```
        <TextView
```

```
            android:id="@+id/tv_progress5"
```

```
            android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:text="00:00"
        android:textColor="@color/white"
        android:textSize="20sp"
        android:fontFamily="@font/hanyi"
        android:textStyle="bold"/>
```

<TextView

```
        android:id="@+id/tv_total5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="04:50"
        android:layout_alignParentRight="true"
        android:textSize="20sp"
        android:textColor="@color/white"
        android:fontFamily="@font/hanyi"
        android:textStyle="bold"/>
```

</RelativeLayout>

<LinearLayout

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_margin="5dp">
```

<Button

```
        android:id="@+id/btn_play5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="开始"
        android:textSize="25sp"
        android:layout_weight="1"
        android:layout_margin="5dp"
        android:textColor="@color/buttonword"
        android:background="@color/buttonbackground"
        android:fontFamily="@font/hanyi"/>
```

<Button

```
        android:id="@+id/btn_pause5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="暫停"
        android:textSize="25sp"
        android:layout_weight="1"
        android:layout_margin="5dp"
        android:textColor="@color/buttonword"
        android:background="@color/buttonbackground"
```

```

        android:fontFamily="@font/hanyi"/>
<Button
    android:id="@+id/btn_continue5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="继续"
    android:textSize="25sp"
    android:layout_weight="1"
    android:layout_margin="5dp"
    android:textColor="@color/buttonword"
    android:background="@color/buttonbackground"
    android:fontFamily="@font/hanyi"/>
<Button
    android:id="@+id/btn_exit5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="退出"
    android:textSize="25sp"
    android:layout_weight="1"
    android:layout_margin="5dp"
    android:textColor="@color/buttonword"
    android:background="@color/buttonbackground"
    android:fontFamily="@font/hanyi"/>

</LinearLayout>

```

</LinearLayout>

in itemlayout.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal">

        <ImageView
            android:id="@+id/iv_image"
            android:layout_width="80dp"
            android:layout_height="80dp"
            android:scaleType="fitXY"/>

```

```

<TextView
    android:id="@+id/tv_name"
    android:layout_width="match_parent"
    android:layout_height="80dp"
    android:textSize="30dp"
    android:textColor="#E3F6FF"
    android:fontFamily="@font/hanyi"
    android:padding="10dp"
    android:layout_marginLeft="20dp"/>

```

```

</LinearLayout>

```

```

</androidx.constraintlayout.widget.ConstraintLayout>

```

设置类代码:

in themes.xml

```

<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Theme.MyApplication9"
parent="Theme.MaterialComponents.DayNight.Bridge">
        <!-- Primary brand color. -->
        <item name="colorPrimary">@color/purple_500</item>
        <item name="colorPrimaryVariant">@color/purple_700</item>
        <item name="colorOnPrimary">@color/white</item>
        <!-- Secondary brand color. -->
        <item name="colorSecondary">@color/teal_200</item>
        <item name="colorSecondaryVariant">@color/teal_700</item>
        <item name="colorOnSecondary">@color/black</item>
        <!-- Status bar color. -->
        <item name="android:statusBarColor"
tools:targetApi="I">?attr/colorPrimaryVariant</item>
        <!-- Customize your theme here. -->
    </style>
</resources>

```

in AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.myapplication9">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"

```



```

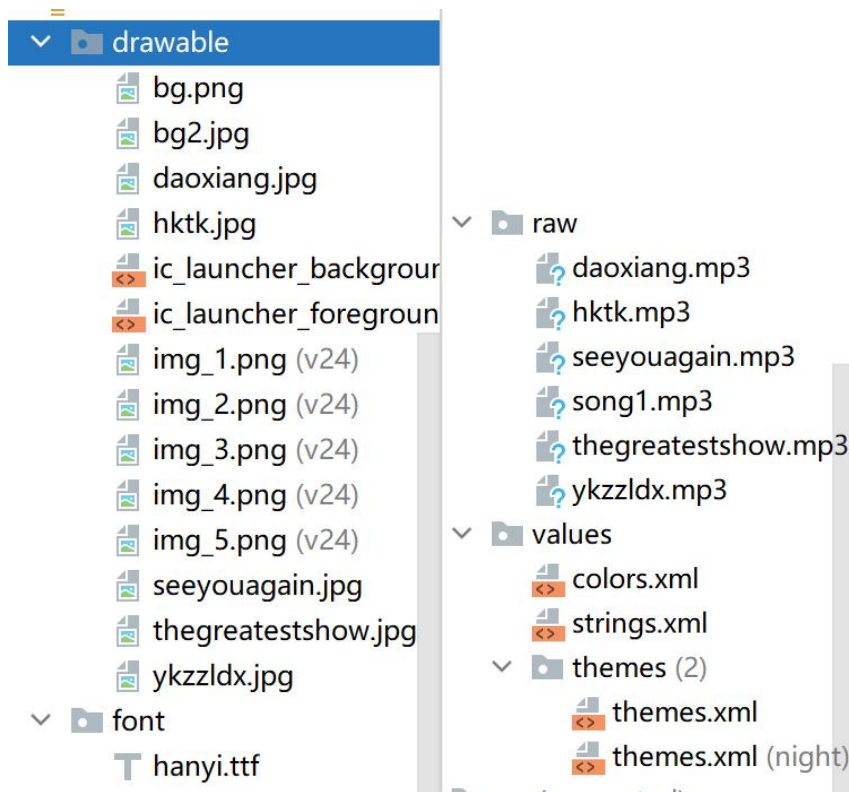
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.MyApplication9">
        <activity
            android:name=".song1"
            android:exported="false" />
        <activity
            android:name=".song2"
            android:exported="false" />
        <activity
            android:name=".song3"
            android:exported="false" />
        <activity
            android:name=".song4"
            android:exported="false" />
        <activity
            android:name=".song5"
            android:exported="false" />
        <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>
        <service android:name=".MusicPlayer1" />
        <service android:name=".MusicPlayer2" />
        <service android:name=".MusicPlayer3" />
        <service android:name=".MusicPlayer4" />
        <service android:name=".MusicPlayer5" />
    </application>

</manifest>

```

内置加入的参数文件：



Activity 类代码:

in MainActivity.java:

package com.example.myapplication9;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import androidx.recyclerview.widget.StaggeredGridLayoutManager;

import android.annotation.SuppressLint;

import android.content.Intent;

import android.graphics.Rect;

import android.os.Bundle;

import android.os.Handler;

import android.provider.ContactsContract;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ImageView;

import android.widget.TextView;

import android.widget.Toast;

public class MainActivity **extends** AppCompatActivity {

public static Handler *handler*;

```

private RecyclerView rv_song;

private int[]
images={R.drawable.daoxiang,R.drawable.ykzzldx,R.drawable.hktk,R.drawable.seeyouagain,R.
drawable.thegreatestshow};
private String[] names={"稻香","夜空中最亮的星","海阔天空","See you again","The
Greatest Show"};
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    rv_song = findViewById(R.id.recycle_view);
    //设置布局管理器
    rv_song.setLayoutManager(new
LinearLayoutManager(getApplicationContext(),LinearLayoutManager.VERTICAL,false));
    //设置适配器
    rv_song.setAdapter(new MyAdapter());
    //设置 itemView 间的间隔
    rv_song.addItemDecoration(new MyDecoration());
}

private class MyAdapter extends RecyclerView.Adapter<MyAdapter.MyHolder> {
    @NonNull
    @Override
    //将 itemlayout 布局转成转成视图，给 MyHolder。
    public MyHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view = View.inflate(getApplicationContext(),R.layout.itemlayout,null);//将
itemlayout 布局转成转成视图
        MyHolder myHolder = new MyHolder(view);//将得到的视图给 MyHolder。
        return myHolder;
    }

    @Override
    //制定每个 holder 里的每个控件的具体内容;
    public void onBindViewHolder(@NonNull MyHolder holder, int position) {
        holder.miv_image.setBackgroundResource(images[position]);
        holder.mtv_name.setText(names[position]);
        //给每个 itemView 设置点击事件，跳转到相应的界面。
        holder.itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

```

```

Intent mintent = null;
switch (position){
    case 0:{
        mintent = new Intent(MainActivity.this,song1.class);
        break;
    }
    case 1:{
        mintent = new Intent(MainActivity.this,song2.class);
        break;
    }
    case 2:{
        mintent = new Intent(MainActivity.this,song3.class);
        break;
    }
    case 3:{
        mintent = new Intent(MainActivity.this,song4.class);
        break;
    }
    case 4:{
        mintent = new Intent(MainActivity.this,song5.class);
        break;
    }
}
startActivity(mintent);
}
});

```

@Override

//给出 item 的数量

```

public int getItemCount() {
    return names.length;
}

```

//声明 MyHolder 里面都有哪些控件， 并和 itemlayout 里面的控件一一对应

```

public class MyHolder extends RecyclerView.ViewHolder {
    ImageView miv_image;
    TextView mtv_name,mtv_info;
    public MyHolder(@NonNull View itemView) {
        super(itemView);
        miv_image = itemView.findViewById(R.id.iv_image);
        mtv_name = itemView.findViewById(R.id.tv_name);
    }
}

```

```

    }
    //自定义 itemView 的间隔
    class MyDecoration extends RecyclerView.ItemDecoration{
        @Override
        public void getItemOffsets(@NonNull Rect outRect, @NonNull View view, @NonNull
RecyclerView parent, @NonNull RecyclerView.State state) {
            super.getItemOffsets(outRect, view, parent, state);
            //            outRect.set(0,0,0,20);
            outRect.set(0,10,0,10);
        }
    }
}

```

in MusicPlayer1.java:

```
package com.example.myapplication9;
```

```

import android.app.Service;
import android.content.Intent;
import android.media.MediaPlayer;
import android.os.Binder;
import android.os.Build;
import android.os.Bundle;
import android.os.IBinder;
import android.os.Message;

```

```
import androidx.annotation.Nullable;
```

```

import java.util.Timer;
import java.util.TimerTask;

```

```

/**
 * 在这里设置音乐播放功能的服务
 *
 */

```

```
public class MusicPlayer1 extends Service {
```

```
    // 设置两个成员变量
```

```
    private MediaPlayer player;//声明一个多媒体对象
```

```
    private Timer timer;//声明一个时钟对象
```

```
    public MusicPlayer1 (){}

```

```
    @Nullable

```

@Override

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

return new MusicControl();*//将 MusicControl() 返回给 onBind () 方法，这样绑定服务的时候，可以把音乐控制器实例化。*

```
}
```

@Override

```
public void onCreate() {
```

```
    super.onCreate();
```

```
    player = new MediaPlayer();//实例化多媒体
```

```
}
```

@Override

```
public void onDestroy() {
```

```
    super.onDestroy();
```

```
    if (player==null) return;
```

```
    if (player.isLooping()) player.stop(); //停止播放音乐
```

```
    player.release();//释放资源
```

```
    player = null;
```

```
}
```

//创建一个内部类 MusicControl，功能是让主程序控制 service 里面的多媒体对象。IBinder 是 Binder 的子类，因此要返回 MusicControl 给 IBinder。

```
class MusicControl extends Binder{
```

```
    public void play() //开始播放
```

```
    {
```

```
        try{
```

```
            player.reset();//重置音乐播放器
```

```
            player = MediaPlayer.create(getApplicationContext(),R.raw.daoxiang); //加载多
```

媒体文件

```
            player.start(); //开始播放音乐
```

```
            addTimer();//添加计时器
```

```
        }catch (Exception exception) //catch 用来处理播放时产生的异常
```

```
            exception.printStackTrace();
```

```
        }
```

```
    }
```

```
    public void pausePlay(){//暂停播放
```

```
        player.pause();
```

```
    }
```

```
    public void continuePlay(){//继续播放
```

```
        player.start();
```

```
    }
```

```
    public void stopPlay(){ //暂停播放
```

```
        player.stop();
```

```
        try {
```

```
            timer.cancel();
```

```

        }catch (Exception e){
            e.printStackTrace();
        }
    }
    public void seekTo(int progress){//定义播放位置播放
        player.seekTo(progress);
    }
}

//添加计时器，计时器是一个多线程的东西，用于设置音乐播放器中的进度条信息
public void addTimer(){
    if (timer == null){
        timer = new Timer();
        TimerTask task = new TimerTask() { //实例化一个计时任务对象
            @Override
            public void run() { //run 就是多线程的一个东西,用于 service 和主线程（即
                MainActivity）之间的通信
                    if (player == null) return; //如果 player 没有实例化，就不执行下面的代码。
                    int duration = player.getDuration();//获取歌曲总长度
                    int currentDuration = player.getCurrentPosition();//获取歌曲当前播放进度
                    //将音乐的总时长、播放时长封装到消息对象中去；
                    Message message = song1.handler.obtainMessage();//在主线程获取一个消
                    息空间

                    Bundle bundle = new Bundle();//定义一个包裹，将歌曲总长度和当前播放长
                    度打包放进去

                    bundle.putInt("duration",duration);
                    bundle.putInt("currentDuration",currentDuration);
                    message.setData(bundle);//将消息包括给 message
                    song1.handler.sendMessage(message);//将消息添加到主线程中
                }
            };
            //开始计时任务后 5ms，执行第一次任务，以后每 500ms 执行一次任务
            timer.schedule(task, 5,1000);
        }
    }
}

in MediaPlayer2.java:
package com.example.myapplication9;

import android.app.Service;
import android.content.Intent;
import android.media.MediaPlayer;
import android.os.Binder;

```

```
import android.os.Bundle;
import android.os.IBinder;
import android.os.Message;
```

```
import androidx.annotation.Nullable;
```

```
import java.util.Timer;
import java.util.TimerTask;
```

```
/**
```

```
 * 在这里设置音乐播放功能的服务
```

```
 *
```

```
 */
```

```
public class MusicPlayer2 extends Service {
```

```
    // 设置两个成员变量
```

```
    private MediaPlayer player; // 声明一个多媒体对象
```

```
    private Timer timer; // 声明一个时钟对象
```

```
    public MusicPlayer2() {}
```

```
    @Nullable
```

```
    @Override
```

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

```
        return new MusicControl(); // 将 MusicControl() 返回给 onBind () 方法，这样绑定服务的时候，可以把音乐控制器实例化。
```

```
    }
```

```
    @Override
```

```
    public void onCreate() {
```

```
        super.onCreate();
```

```
        player = new MediaPlayer(); // 实例化多媒体
```

```
    }
```

```
    @Override
```

```
    public void onDestroy() {
```

```
        super.onDestroy();
```

```
        if (player == null) return;
```

```
        if (player.isLooping()) player.stop(); // 停止播放音乐
```

```
        player.release(); // 释放资源
```

```
        player = null;
```

```
    }
```

```
    // 创建一个内部类 MusicControl，功能是让主程序控制 service 里面的多媒体对象。IBinder 是
```


Binder 的子类，因此要返回 MusicControl 给 IBinder。

```
class MusicControl extends Binder{
    public void play() { // 开始播放
        try{
            player.reset(); // 重置音乐播放器
            player = MediaPlayer.create(getApplicationContext(), R.raw.ykzzlhx); // 加载多媒体文件

            player.start(); // 开始播放音乐
            addTimer(); // 添加计时器
        } catch (Exception exception) { // catch 用来处理播放时产生的异常
            exception.printStackTrace();
        }
    }
    public void pausePlay() { // 暂停播放
        player.pause();
    }
    public void continuePlay() { // 继续播放
        player.start();
    }
    public void stopPlay() { // 暂停播放
        player.stop();
        try {
            timer.cancel();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    public void seekTo(int progress) { // 定义播放位置播放
        player.seekTo(progress);
    }
}
```

// 添加计时器，计时器是一个多线程的东西，用于设置音乐播放器中的进度条信息

```
public void addTimer(){
    if (timer == null){
        timer = new Timer();
        TimerTask task = new TimerTask() { // 实例化一个计时任务对象
            @Override
            public void run() { // run 就是多线程的一个东西，用于 service 和主线程（即 MainActivity）之间的通信
                if (player == null) return; // 如果 player 没有实例化，就不执行下面的代码。
                int duration = player.getDuration(); // 获取歌曲总长度
                int currentDuration = player.getCurrentPosition(); // 获取歌曲当前播放进度
                // 将音乐的总时长、播放时长封装到消息对象中去；
            }
        };
        timer.schedule(task, 0);
    }
}
```

```

        Message message = song2.handler.obtainMessage();//在主线程获取一个消
        息空间
        Bundle bundle = new Bundle();//定义一个包裹，将歌曲总长度和当前播放长
        度打包放进去
        bundle.putInt("duration",duration);
        bundle.putInt("currentDuration",currentDuration);
        message.setData(bundle);//将消息包括给 message
        song2.handler.sendMessage(message);//将消息添加到主线程中
    }
};
//开始计时任务后 5ms，执行第一次任务，以后每 500ms 执行一次任务
timer.schedule(task, 5,1000);
}
}
}

```

in MediaPlayer3.java:

```
package com.example.myapplication9;
```

```

import android.app.Service;
import android.content.Intent;
import android.media.MediaPlayer;
import android.os.Binder;
import android.os.Bundle;
import android.os.IBinder;
import android.os.Message;

```

```
import androidx.annotation.Nullable;
```

```

import java.util.Timer;
import java.util.TimerTask;

```

```

/**
 * 在这里设置音乐播放功能的服务
 *
 */

```

```
public class MediaPlayer3 extends Service {
```

```

    // 设置两个成员变量
    private MediaPlayer player;//声明一个多媒体对象
    private Timer timer;//声明一个时钟对象

```

```
public MediaPlayer3(){}
```

```

@Nullable
@Override
public IBinder onBind(Intent intent) {
    return new MusicControl();//将MusicControl()返回给onBind () 方法，这样绑定服务的时候，可以把音乐控制器实例化。
}

```

```

@Override
public void onCreate() {
    super.onCreate();
    player = new MediaPlayer();//实例化多媒体
}

```

```

@Override
public void onDestroy() {
    super.onDestroy();
    if (player==null) return;
    if (player.isLooping()) player.stop(); //停止播放音乐
    player.release();//释放资源
    player = null;
}

```

//创建一个内部类 MusicControl，功能是让主程序控制 service 里面的多媒体对象。IBinder 是 Binder 的子类，因此要返回 MusicControl 给 IBinder。

```

class MusicControl extends Binder{
    public void play() { //开始播放
        try{
            player.reset();//重置音乐播放器
            player = MediaPlayer.create(getApplicationContext(),R.raw.hktk); //加载多媒体文件

            player.start(); //开始播放音乐
            addTimer();//添加计时器
        }catch (Exception exception) { //catch 用来处理播放时产生的异常
            exception.printStackTrace();
        }
    }
    public void pausePlay(){ //暂停播放
        player.pause();
    }
    public void continuePlay(){ //继续播放
        player.start();
    }
    public void stopPlay(){ //暂停播放
        player.stop();
    }
}

```

```

        try {
            timer.cancel();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public void seekTo(int progress){//定义播放位置播放
        player.seekTo(progress);
    }
}

//添加计时器，计时器是一个多线程的东西，用于设置音乐播放器中的进度条信息
public void addTimer(){
    if (timer == null){
        timer = new Timer();
        TimerTask task = new TimerTask() { //实例化一个计时任务对象
            @Override
            public void run() { //run 就是多线程的一个东西,用于 service 和主线程（即
                MainActivity）之间的通信

                if (player == null) return; //如果 player 没有实例化，就不执行下面的代码。
                int duration = player.getDuration();//获取歌曲总长度
                int currentDuration = player.getCurrentPosition();//获取歌曲当前播放进度
                //将音乐的总时长、播放时长封装到消息对象中去；
                Message message = song3.handler.obtainMessage();//在主线程获取一个消
                息空间

                Bundle bundle = new Bundle();//定义一个包裹，将歌曲总长度和当前播放长
                度打包放进去

                bundle.putInt("duration",duration);
                bundle.putInt("currentDuration",currentDuration);
                message.setData(bundle);//将消息包括给 message
                song3.handler.sendMessage(message);//将消息添加到主线程中
            }
        };
        //开始计时任务后 5ms，执行第一次任务，以后每 500ms 执行一次任务
        timer.schedule(task, 5,1000);
    }
}

}

in MediaPlayer4.java:
package com.example.myapplication9;

import android.app.Service;
import android.content.Intent;

```

```
import android.media.MediaPlayer;
import android.os.Binder;
import android.os.Bundle;
import android.os.IBinder;
import android.os.Message;
```

```
import androidx.annotation.Nullable;
```

```
import java.util.Timer;
import java.util.TimerTask;
```

```
/**
```

```
 * 在这里设置音乐播放功能的服务
```

```
 *
```

```
 */
```

```
public class MusicPlayer4 extends Service {
```

```
    // 设置两个成员变量
```

```
    private MediaPlayer player; // 声明一个多媒体对象
```

```
    private Timer timer; // 声明一个时钟对象
```

```
    public MusicPlayer4() {}
```

```
    @Nullable
```

```
    @Override
```

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

return new MusicControl(); // 将 MusicControl() 返回给 onBind () 方法，这样绑定服务的时候，可以把音乐控制器实例化。

```
    }
```

```
    @Override
```

```
    public void onCreate() {
```

```
        super.onCreate();
```

```
        player = new MediaPlayer(); // 实例化多媒体
```

```
    }
```

```
    @Override
```

```
    public void onDestroy() {
```

```
        super.onDestroy();
```

```
        if (player==null) return;
```

```
        if (player.isLooping()) player.stop(); // 停止播放音乐
```

```
        player.release(); // 释放资源
```

```
        player = null;
```

```
}
```

//创建一个内部类 MusicControl，功能是让主程序控制 service 里面的多媒体对象。IBinder 是 Binder 的子类，因此要返回 MusicControl 给 IBinder。

```
class MusicControl extends Binder{
```

```
    public void play() {//开始播放
```

```
        try{
```

```
            player.reset();//重置音乐播放器
```

```
            player = MediaPlayer.create(getApplicationContext(),R.raw.seeyouagain); //加载多媒体文件
```

```
            player.start(); //开始播放音乐
```

```
            addTimer();//添加计时器
```

```
        }catch (Exception exception) {//catch 用来处理播放时产生的异常
```

```
            exception.printStackTrace();
```

```
        }
```

```
    }
```

```
    public void pausePlay(){//暂停播放
```

```
        player.pause();
```

```
    }
```

```
    public void continuePlay(){//继续播放
```

```
        player.start();
```

```
    }
```

```
    public void stopPlay(){ //暂停播放
```

```
        player.stop();
```

```
        try {
```

```
            timer.cancel();
```

```
        }catch (Exception e){
```

```
            e.printStackTrace();
```

```
        }
```

```
    }
```

```
    public void seekTo(int progress){//定义播放位置播放
```

```
        player.seekTo(progress);
```

```
    }
```

```
}
```

//添加计时器，计时器是一个多线程的东西，用于设置音乐播放器中的进度条信息

```
public void addTimer(){
```

```
    if (timer == null){
```

```
        timer = new Timer();
```

```
        TimerTask task = new TimerTask() {//实例化一个计时任务对象
```

```
            @Override
```

```
            public void run() { //run 就是多线程的一个东西,用于 service 和主线程（即 MainActivity）之间的通信
```

```
                if (player == null) return; //如果 player 没有实例化，就不执行下面的代码。
```

```
                int duration = player.getDuration();//获取歌曲总长度
```

```

        int currentDuration = player.getCurrentPosition();//获取歌曲当前播放进度
        //将音乐的总时长、播放时长封装到消息对象中去;
        Message message = song4.handler.obtainMessage();//在主线程获取一个消息空间

        Bundle bundle = new Bundle();//定义一个包裹，将歌曲总长度和当前播放长度打包放进去

        bundle.putInt("duration",duration);
        bundle.putInt("currentDuration",currentDuration);
        message.setData(bundle);//将消息包括给 message
        song4.handler.sendMessage(message);//将消息添加到主线程中
    }
};
//开始计时任务后 5ms，执行第一次任务，以后每 500ms 执行一次任务
timer.schedule(task, 5,1000);
}
}
}

```

in MediaPlayer5.java:

```
package com.example.myapplication9;
```

```

import android.app.Service;
import android.content.Intent;
import android.media.MediaPlayer;
import android.os.Binder;
import android.os.Bundle;
import android.os.IBinder;
import android.os.Message;

```

```
import androidx.annotation.Nullable;
```

```

import java.util.Timer;
import java.util.TimerTask;

```

```
/**
```

```
 * 在这里设置音乐播放功能的服务
```

```
 *
```

```
*/
```

```
public class MediaPlayer5 extends Service {
```

```
    // 设置两个成员变量
```

```
    private MediaPlayer player;//声明一个多媒体对象
```

```
    private Timer timer;//声明一个时钟对象
```

```
public MusicPlayer5(){}
```

```
@Nullable
```

```
@Override
```

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

return new MusicControl();*//将 MusicControl() 返回给 onBind () 方法，这样绑定服务的时候，可以把音乐控制器实例化。*

```
}
```

```
@Override
```

```
public void onCreate() {
```

```
    super.onCreate();
```

```
    player = new MediaPlayer();//实例化多媒体
```

```
}
```

```
@Override
```

```
public void onDestroy() {
```

```
    super.onDestroy();
```

```
    if (player==null) return;
```

```
    if (player.isLooping()) player.stop();//停止播放音乐
```

```
    player.release();//释放资源
```

```
    player = null;
```

```
}
```

//创建一个内部类 MusicControl，功能是让主程序控制 service 里面的多媒体对象。IBinder 是 Binder 的子类，因此要返回 MusicControl 给 IBinder。

```
class MusicControl extends Binder{
```

```
    public void play() {//开始播放
```

```
        try{
```

```
            player.reset();//重置音乐播放器
```

```
            player = MediaPlayer.create(getApplicationContext(),R.raw.thegreatestshow);
```

//加载多媒体文件

```
            player.start();//开始播放音乐
```

```
            addTimer();//添加计时器
```

```
        }catch (Exception exception) {//catch 用来处理播放时产生的异常  
            exception.printStackTrace();
```

```
        }
```

```
    }
```

```
    public void pausePlay(){//暂停播放
```

```
        player.pause();
```

```
    }
```

```
    public void continuePlay(){//继续播放
```

```
        player.start();
```

```
    }
```



```

public void stopPlay(){ //暂停播放
    player.stop();
    try {
        timer.cancel();
    }catch (Exception e){
        e.printStackTrace();
    }
}
public void seekTo(int progress){ //定义播放位置播放
    player.seekTo(progress);
}
}

```

//添加计时器，计时器是一个多线程的东西，用于设置音乐播放器中的进度条信息

```

public void addTimer(){
    if (timer == null){
        timer = new Timer();
        TimerTask task = new TimerTask() { //实例化一个计时任务对象
            @Override
            public void run() { //run 就是多线程的一个东西,用于 service 和主线程 (即
                MainActivity) 之间的通信
                    if (player == null) return; //如果 player 没有实例化，就不执行下面的代码。
                    int duration = player.getDuration(); //获取歌曲总长度
                    int currentDuration = player.getCurrentPosition(); //获取歌曲当前播放进度
                    //将音乐的总时长、播放时长封装到消息对象中去；
                    Message message = song5.handler.obtainMessage(); //在主线程获取一个消
                    息空间
                    Bundle bundle = new Bundle(); //定义一个包裹，将歌曲总长度和当前播放长
                    度打包放进去
                    bundle.putInt("duration", duration);
                    bundle.putInt("currentDuration", currentDuration);
                    message.setData(bundle); //将消息包括给 message
                    song5.handler.sendMessage(message); //将消息添加到主线程中
                }
            };
            //开始计时任务后 5ms，执行第一次任务，以后每 500ms 执行一次任务
            timer.schedule(task, 5, 1000);
        }
    }
}

```

Service 类代码：

in Song1.java:

```

package com.example.myapplication9;

```

```

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.animation.ObjectAnimator;
import android.content.ComponentName;
import android.content.Intent;
import android.content.ServiceConnection;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.os.Handler;
import android.os.HandlerThread;
import android.os.IBinder;
import android.os.Looper;
import android.os.Message;
import android.view.View;
import android.view.animation.LinearInterpolator;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.SeekBar;
import android.widget.TextView;

import java.util.Timer;

public class song1 extends AppCompatActivity{
    private ImageView iv_cover;
    private static SeekBar sb;
    private static TextView tv_progress,tv_total;
    private Button btn_play,btn_pause,btn_continue,btn_exit;

    private ObjectAnimator animator; //声明一个动画组件 ObjectAnimator

    private MediaPlayer1.MusicControl control; //声明 MusicService 中的音乐控制器

    private ServiceConnection connection = new ServiceConnection() { //声明服务连接
        @Override
        public void onServiceConnected(ComponentName componentName, IBinder iBinder) {
            control = (MediaPlayer1.MusicControl) iBinder; //实例化音乐控制对象，即 control。
        }
        @Override
        public void onServiceDisconnected(ComponentName componentName) {

        }
    };
};

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_song1);
    init();
}

public void init(){
    iv_cover = findViewById(R.id.iv_cover1);
    sb = findViewById(R.id.sb1);
    tv_progress = findViewById(R.id.tv_progress1);
    tv_total = findViewById(R.id.tv_total1);

    btn_play = findViewById(R.id.btn_play1);
    btn_pause = findViewById(R.id.btn_pause1);
    btn_continue = findViewById(R.id.btn_continue1);
    btn_exit = findViewById(R.id.btn_exit1);

    OnClickListener monclick = new OnClickListener();
    btn_play.setOnClickListener(monclick);
    btn_pause.setOnClickListener(monclick);
    btn_continue.setOnClickListener(monclick);
    btn_exit.setOnClickListener(monclick);

    //执行动画的对象是 iv_cover, // 动画效果是 0-360° 旋转 (用的是浮点数, 所以加个f)。
    animator = ObjectAnimator.ofFloat(iv_cover,"rotation",0.0f,360.0f);
    animator.setDuration(10000); //旋转一周的时长, 单位是毫秒, 此处设置了 10s
    animator.setInterpolator(new LinearInterpolator()); //设置匀速转动
    animator.setRepeatCount(-1); //设置循环, 此处设置的是无限循环。如果是正值, 意味着转动多少圈。

    //声明一个意图, 该意图进行服务的启动, 意思是将 MusicService 里面的服务要传到主程序这里来。
    Intent mintent = new Intent(song1.this,MusicPlayer1.class);
    bindService(mintent,connection,BIND_AUTO_CREATE); //建立意图中 MainActivity 与 MusicService 两对象的服务连接

    seekBarListener msbListener = new seekBarListener();
    sb.setOnSeekBarChangeListener(msbListener);
}

// 设置播放、暂停、继续和退出按钮的监听 (或点击) 事件
class OnClickListener implements View.OnClickListener{

```

```

@Override
public void onClick(View view) {
    switch (view.getId()){
        case R.id.btn_play1:
            //播放音乐
            control.play();
            //光盘开始转
            animator.start();
            break;
        case R.id.btn_pause1:
            //停止播放音乐
            control.pausePlay();
            //光盘停止转
            animator.pause();
            break;
        case R.id.btn_continue1:
            //继续播放音乐
            control.continuePlay();
            //光盘继续转
            animator.resume();
            break;
        case R.id.btn_exit1:
            finish();
            break;
    }
}
}

```

```

@Override
protected void onDestroy() {
    control.stopPlay();
    unbindService(connection);
    super.onDestroy();
}

```

//Handler 主要用于异步消息的处理，在这里是处理子线程 MusicService 传来的消息

```

public static Handler handler = new Handler(Looper.getMainLooper()){

```

```

@Override
public void handleMessage(@NonNull Message msg) {
    //super.handleMessage(msg);
    Bundle bundle = msg.getData();
    int duration = bundle.getInt("duration");//把音乐时长放在 bundle 里

```

bundle 里

```
int currentDuration = bundle.getInt("currentDuration");//把音乐当前播放时长放在
```

```
sb.setMax(duration);  
sb.setProgress(currentDuration);
```

//显示总时长

```
int minite = duration / 1000 / 60;  
int second = duration / 1000 % 60;  
String strMinite = "";  
String strSecond = "";  
if (minite < 10){  
    strMinite = "0" + minite;  
}else {  
    strMinite = minite + "";  
}  
if (second < 10){  
    strSecond = "0" + second;  
}else {  
    strSecond = second + "";  
}  
tv_total.setText(strMinite + ":" + strSecond);
```

//显示播放时长

```
minite = currentDuration / 1000 / 60;  
second = currentDuration / 1000 % 60;  
  
if (minite < 10){  
    strMinite = "0" + minite;  
}else {  
    strMinite = minite + "";  
}  
if (second < 10){  
    strSecond = "0" + second;  
}else {  
    strSecond = second + "";  
}  
tv_progress.setText(strMinite + ":" + strSecond);  
}  
};
```

//给进度条设置监听

```
class seekBarListener implements SeekBar.OnSeekBarChangeListener {
```

@Override

//进度条行进过程的监听

```
public void onProgressChanged(SeekBar seekBar, int i, boolean b) {  
    if (i == seekBar.getMax()){  
        animator.pause();  
    }  
    if (b){//判断是否来自用户  
        control.seekTo(i);  
    }  
}
```

@Override

//用户开始滑动进度条的监听

```
public void onStartTrackingTouch(SeekBar seekBar) {  
    control.pausePlay();  
    animator.pause();  
}
```

@Override

//用户停止滑动进度条的监听

```
public void onStopTrackingTouch(SeekBar seekBar) {  
    control.continuePlay();  
    animator.resume();  
}  
}
```

}

in Song2.java:

```
package com.example.myapplication9;
```

```
import android.animation.ObjectAnimator;  
import android.content.ComponentName;  
import android.content.Intent;  
import android.content.ServiceConnection;  
import android.os.Bundle;  
import android.os.Handler;  
import android.os.IBinder;  
import android.os.Looper;  
import android.os.Message;  
import android.view.View;  
import android.view.animation.LinearInterpolator;  
import android.widget.Button;  
import android.widget.ImageView;
```

```

import android.widget.SeekBar;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

public class song2 extends AppCompatActivity{
    private ImageView iv_cover;
    private static SeekBar sb;
    private static TextView tv_progress,tv_total;
    private Button btn_play,btn_pause,btn_continue,btn_exit;

    private ObjectAnimator animator; //声明一个动画组件 ObjectAnimator

    private MediaPlayer2.MusicControl control; //声明 MusicService 中的音乐控制器

    private ServiceConnection connection = new ServiceConnection() { //声明服务连接
        @Override
        public void onServiceConnected(ComponentName componentName, IBinder iBinder) {
            control = (MediaPlayer2.MusicControl) iBinder; //实例化音乐控制对象，即 control。
        }
        @Override
        public void onServiceDisconnected(ComponentName componentName) {

        }
    };

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_song2);
        init();
    }
    public void init(){
        iv_cover = findViewById(R.id.iv_cover2);
        sb = findViewById(R.id.sb2);
        tv_progress = findViewById(R.id.tv_progress2);
        tv_total = findViewById(R.id.tv_total2);

        btn_play = findViewById(R.id.btn_play2);
        btn_pause = findViewById(R.id.btn_pause2);
        btn_continue = findViewById(R.id.btn_continue2);
        btn_exit = findViewById(R.id.btn_exit2);
    }

```

```
OnClick monclick = new OnClick();
btn_play.setOnClickListener(monclick);
btn_pause.setOnClickListener(monclick);
btn_continue.setOnClickListener(monclick);
btn_exit.setOnClickListener(monclick);
```

//执行动画的对象是 iv_cover, // 动画效果是 0-360° 旋转 (用的是浮点数, 所以加个f)。

```
animator = ObjectAnimator.ofFloat(iv_cover,"rotation",0.0f,360.0f);
animator.setDuration(10000); //旋转一周的时长, 单位是毫秒, 此处设置了 10s
animator.setInterpolator(new LinearInterpolator()); //设置匀速转动
animator.setRepeatCount(-1); //设置循环, 此处设置的是无限循环。如果是正值, 意味着转动多少圈。
```

//声明一个意图, 该意图进行服务的启动, 意思是将 MusicService 里面的服务要传到主程序这里来。

```
Intent mintent = new Intent(song2.this,MusicPlayer2.class);
bindService(mintent,connection,BIND_AUTO_CREATE); //建立意图中 MainActivity 与 MusicService 两对象的服务连接
```

```
seekBarListener msbListener = new seekBarListener();
sb.setOnSeekBarChangeListener(msbListener);
```

```
}
```

// 设置播放、暂停、继续和退出按钮的监听 (或点击) 事件

```
class OnClick implements View.OnClickListener{
```

```
@Override
```

```
public void onClick(View view) {
    switch (view.getId()){
        case R.id.btn_play2:
            //播放音乐
            control.play();
            //光盘开始转
            animator.start();
            break;
        case R.id.btn_pause2:
            //停止播放音乐
            control.pausePlay();
            //光盘停止转
            animator.pause();
            break;
        case R.id.btn_continue2:
            //继续播放音乐
```



```

        control.continuePlay();
        //光盘继续转
        animator.resume();
        break;
    case R.id.btn_exit2:
        finish();
        break;
    }
}
}
}

```

```

@Override
protected void onDestroy() {
    control.stopPlay();
    unbindService(connection);
    super.onDestroy();
}

```

//Handler 主要用于异步消息的处理，在这里是处理子线程 MusicService 传来的消息

```

public static Handler handler = new Handler(Looper.getMainLooper()){

```

```

    @Override
    public void handleMessage(@NonNull Message msg) {
        //super.handleMessage(msg);
        Bundle bundle = msg.getData();
        int duration = bundle.getInt("duration");//把音乐时长放在 bundle 里
        int currentDuration = bundle.getInt("currentDuration");//把音乐当前播放时长放在
        bundle 里

```

```

        sb.setMax(duration);
        sb.setProgress(currentDuration);

```

//显示总时长

```

        int minite = duration / 1000 / 60;
        int second = duration / 1000 % 60;
        String strMinite = "";
        String strSecond = "";
        if (minite < 10){
            strMinite = "0" + minite;
        }else {
            strMinite = minite + "";
        }
        if (second < 10){

```

```

        strSecond = "0" + second;
    }else {
        strSecond = second + "";
    }
    tv_total.setText(strMinite + ":" + strSecond);

    //显示播放时长
    minite = currentDuration / 1000 / 60;
    second = currentDuration / 1000 % 60;

    if (minite < 10){
        strMinite = "0" + minite;
    }else {
        strMinite = minite + "";
    }
    if (second < 10){
        strSecond = "0" + second;
    }else {
        strSecond = second + "";
    }
    tv_progress.setText(strMinite + ":" + strSecond);
}

};

//给进度条设置监听
class seekBarListener implements SeekBar.OnSeekBarChangeListener {
    @Override
    //进度条行进过程的监听
    public void onProgressChanged(SeekBar seekBar, int i, boolean b) {
        if (i == seekBar.getMax()){
            animator.pause();
        }
        if (b){//判断是否来自用户
            control.seekTo(i);
        }
    }

    @Override
    //用户开始滑动进度条的监听
    public void onStartTrackingTouch(SeekBar seekBar) {
        control.pausePlay();
        animator.pause();
    }
}

```

```

        @Override
        //用户停止滑动进度条的监听
        public void onStopTrackingTouch(SeekBar seekBar) {
            control.continuePlay();
            animator.resume();
        }
    }
}

```

in Song3.java:

```
package com.example.myapplication9;
```

```

import android.animation.ObjectAnimator;
import android.content.ComponentName;
import android.content.Intent;
import android.content.ServiceConnection;
import android.os.Bundle;
import android.os.Handler;
import android.os.IBinder;
import android.os.Looper;
import android.os.Message;
import android.view.View;
import android.view.animation.LinearInterpolator;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.SeekBar;
import android.widget.TextView;

```

```

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

```

```

public class song3 extends AppCompatActivity{
    private ImageView iv_cover;
    private static SeekBar sb;
    private static TextView tv_progress,tv_total;
    private Button btn_play,btn_pause,btn_continue,btn_exit;

    private ObjectAnimator animator; //声明一个动画组件 ObjectAnimator

    private MediaPlayer3.MusicControl control; //声明 MusicService 中的音乐控制器

    private ServiceConnection connection = new ServiceConnection() { //声明服务连接

```

```

@Override
public void onServiceConnected(ComponentName componentName, IBinder iBinder) {
    control = (MediaPlayer3.MusicControl) iBinder; //实例化音乐控制对象，即 control。
}
@Override
public void onServiceDisconnected(ComponentName componentName) {

}

};

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_song3);
    init();
}
public void init(){
    iv_cover = findViewById(R.id.iv_cover3);
    sb = findViewById(R.id.sb3);
    tv_progress = findViewById(R.id.tv_progress3);
    tv_total = findViewById(R.id.tv_total3);

    btn_play = findViewById(R.id.btn_play3);
    btn_pause = findViewById(R.id.btn_pause3);
    btn_continue = findViewById(R.id.btn_continue3);
    btn_exit = findViewById(R.id.btn_exit3);

    OnClickListener monclick = new OnClickListener();
    btn_play.setOnClickListener(monclick);
    btn_pause.setOnClickListener(monclick);
    btn_continue.setOnClickListener(monclick);
    btn_exit.setOnClickListener(monclick);

    //执行动画的对象是 iv_cover, // 动画效果是 0-360° 旋转（用的是浮点数，所以加个f）。
    animator = ObjectAnimator.ofFloat(iv_cover, "rotation", 0.0f, 360.0f);
    animator.setDuration(10000); //旋转一周的时长，单位是毫秒，此处设置了 10s
    animator.setInterpolator(new LinearInterpolator()); //设置匀速转动
    animator.setRepeatCount(-1); //设置循环，此处设置的是无限循环。如果是正值，意味着转动多少圈。

```

//声明一个意图，该意图进行服务的启动，意思是将 **MusicService** 里面的服务要传到主程序这里来。

```
Intent mintent = new Intent(song3.this, MediaPlayer3.class);
```

bindService(intent,connection,BIND_AUTO_CREATE);//建立意图中 MainActivity 与
MusicService 两对象的服务连接

```
seekBarListener msbListener = new seekBarListener();  
sb.setOnSeekBarChangeListener(msbListener);
```

```
}
```

// 设置播放、暂停、继续和退出按钮的监听（或点击）事件

```
class OnClick implements View.OnClickListener{
```

```
@Override
```

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

```
    switch (view.getId()){
```

```
        case R.id.btn_play3:
```

```
            //播放音乐
```

```
            control.play();
```

```
            //光盘开始转
```

```
            animator.start();
```

```
            break;
```

```
        case R.id.btn_pause3:
```

```
            //停止播放音乐
```

```
            control.pausePlay();
```

```
            //光盘停止转
```

```
            animator.pause();
```

```
            break;
```

```
        case R.id.btn_continue3:
```

```
            //继续播放音乐
```

```
            control.continuePlay();
```

```
            //光盘继续转
```

```
            animator.resume();
```

```
            break;
```

```
        case R.id.btn_exit3:
```

```
            finish();
```

```
            break;
```

```
    }
```

```
}
```

```
}
```

```
@Override
```

```
protected void onDestroy() {
```

```
    control.stopPlay();
```

```
    unbindService(connection);
```

```
    super.onDestroy();
```

```
}
```

//Handler 主要用于异步消息的处理，在这里是处理子线程 MusicService 传来的消息

```
public static Handler handler = new Handler(Looper.getMainLooper()){
```

```
    @Override
```

```
    public void handleMessage(@NonNull Message msg) {
```

```
        //super.handleMessage(msg);
```

```
        Bundle bundle = msg.getData();
```

```
        int duration = bundle.getInt("duration");//把音乐时长放在 bundle 里
```

```
        int currentDuration = bundle.getInt("currentDuration");//把音乐当前播放时长放在 bundle 里
```

```
        sb.setMax(duration);
```

```
        sb.setProgress(currentDuration);
```

```
        //显示总时长
```

```
        int minite = duration / 1000 / 60;
```

```
        int second = duration / 1000 % 60;
```

```
        String strMinite = "";
```

```
        String strSecond = "";
```

```
        if (minite < 10){
```

```
            strMinite = "0" + minite;
```

```
        }else {
```

```
            strMinite = minite + "";
```

```
        }
```

```
        if (second < 10){
```

```
            strSecond = "0" + second;
```

```
        }else {
```

```
            strSecond = second + "";
```

```
        }
```

```
        tv_total.setText(strMinite + ":" + strSecond);
```

```
        //显示播放时长
```

```
        minite = currentDuration / 1000 / 60;
```

```
        second = currentDuration / 1000 % 60;
```

```
        if (minite < 10){
```

```
            strMinite = "0" + minite;
```

```
        }else {
```

```
            strMinite = minite + "";
```

```
        }
```

```
        if (second < 10){
```

```

        strSecond = "0" + second;
    }else {
        strSecond = second + "";
    }
    tv_progress.setText(strMinitime + ":" + strSecond);
}
};

```

//给进度条设置监听

```

class seekBarListener implements SeekBar.OnSeekBarChangeListener {
    @Override
    //进度条行进过程的监听
    public void onProgressChanged(SeekBar seekBar, int i, boolean b) {
        if (i == seekBar.getMax()){
            animator.pause();
        }
        if (b){//判断是否来自用户
            control.seekTo(i);
        }
    }

    @Override
    //用户开始滑动进度条的监听
    public void onStartTrackingTouch(SeekBar seekBar) {
        control.pausePlay();
        animator.pause();
    }

    @Override
    //用户停止滑动进度条的监听
    public void onStopTrackingTouch(SeekBar seekBar) {
        control.continuePlay();
        animator.resume();
    }
}

```

}

in Song4.java:

```
package com.example.myapplication9;
```

```
import android.animation.ObjectAnimator;
```

```
import android.content.ComponentName;
```

```
import android.content.Intent;
```

```

import android.content.ServiceConnection;
import android.os.Bundle;
import android.os.Handler;
import android.os.IBinder;
import android.os.Looper;
import android.os.Message;
import android.view.View;
import android.view.animation.LinearInterpolator;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.SeekBar;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

public class song4 extends AppCompatActivity{
    private ImageView iv_cover;
    private static SeekBar sb;
    private static TextView tv_progress,tv_total;
    private Button btn_play,btn_pause,btn_continue,btn_exit;

    private ObjectAnimator animator; //声明一个动画组件 ObjectAnimator

    private MediaPlayer4.MusicControl control; //声明 MusicService 中的音乐控制器

    private ServiceConnection connection = new ServiceConnection() { //声明服务连接
        @Override
        public void onServiceConnected(ComponentName componentName, IBinder iBinder) {
            control = (MediaPlayer4.MusicControl) iBinder; //实例化音乐控制对象，即 control。
        }
        @Override
        public void onServiceDisconnected(ComponentName componentName) {

        }
    };

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_song4);
        init();
    }

```



```

public void init(){
    iv_cover = findViewById(R.id.iv_cover4);
    sb = findViewById(R.id.sb4);
    tv_progress = findViewById(R.id.tv_progress4);
    tv_total = findViewById(R.id.tv_total4);

    btn_play = findViewById(R.id.btn_play4);
    btn_pause = findViewById(R.id.btn_pause4);
    btn_continue = findViewById(R.id.btn_continue4);
    btn_exit = findViewById(R.id.btn_exit4);

```

```

OnClick monclick = new OnClick();
btn_play.setOnClickListener(monclick);
btn_pause.setOnClickListener(monclick);
btn_continue.setOnClickListener(monclick);
btn_exit.setOnClickListener(monclick);

```

//执行动画的对象是 iv_cover, // 动画效果是 0-360° 旋转 (用的是浮点数, 所以加个f)。

```

animator = ObjectAnimator.ofFloat(iv_cover,"rotation",0.0f,360.0f);

```

```

animator.setDuration(10000); //旋转一周的时长, 单位是毫秒, 此处设置了 10s

```

```

animator.setInterpolator(new LinearInterpolator()); //设置匀速转动

```

animator.setRepeatCount(-1); //设置循环, 此处设置的是无限循环。如果是正值, 意味着转动多少圈。

//声明一个意图, 该意图进行服务的启动, 意思是将 MusicService 里面的服务要传到主程序这里来。

```

Intent mintent = new Intent(song4.this,MusicPlayer4.class);

```

bindService(mintent,connection,BIND_AUTO_CREATE); //建立意图中 MainActivity 与 MusicService 两对象的服务连接

```

seekBarListener msbListener = new seekBarListener();
sb.setOnSeekBarChangeListener(msbListener);

```

```

}

```

// 设置播放、暂停、继续和退出按钮的监听 (或点击) 事件

```

class OnClick implements View.OnClickListener{

```

```

@Override

```

```

public void onClick(View view) {

```

```

    switch (view.getId()){

```

```

        case R.id.btn_play4:

```

//播放音乐

```

        control.play();

```

//光盘开始转

```

        animator.start();
        break;
    case R.id.btn_pause4:
        //停止播放音乐
        control.pausePlay();
        //光盘停止转
        animator.pause();
        break;
    case R.id.btn_continue4:
        //继续播放音乐
        control.continuePlay();
        //光盘继续转
        animator.resume();
        break;
    case R.id.btn_exit4:
        finish();
        break;
    }
}
}
}

```

```

@Override
protected void onDestroy() {
    control.stopPlay();
    unbindService(connection);
    super.onDestroy();
}

```

//Handler 主要用于异步消息的处理，在这里是处理子线程 MusicService 传来的消息

```

public static Handler handler = new Handler(Looper.getMainLooper()){

```

```

    @Override
    public void handleMessage(@NonNull Message msg) {
        //super.handleMessage(msg);
        Bundle bundle = msg.getData();
        int duration = bundle.getInt("duration");//把音乐时长放在 bundle 里
        int currentDuration = bundle.getInt("currentDuration");//把音乐当前播放时长放在
        bundle 里

        sb.setMax(duration);
        sb.setProgress(currentDuration);

        //显示总时长
    }
}

```

```

    int minite = duration / 1000 / 60;
    int second = duration / 1000 % 60;
    String strMinite = "";
    String strSecond = "";
    if (minite < 10){
        strMinite = "0" + minite;
    }else {
        strMinite = minite + "";
    }
    if (second < 10){
        strSecond = "0" + second;
    }else {
        strSecond = second + "";
    }
    tv_total.setText(strMinite + ":" + strSecond);

    //显示播放时长
    minite = currentDuration / 1000 / 60;
    second = currentDuration / 1000 % 60;

    if (minite < 10){
        strMinite = "0" + minite;
    }else {
        strMinite = minite + "";
    }
    if (second < 10){
        strSecond = "0" + second;
    }else {
        strSecond = second + "";
    }
    tv_progress.setText(strMinite + ":" + strSecond);
}

};

//给进度条设置监听
class seekBarListener implements SeekBar.OnSeekBarChangeListener {
    @Override
    //进度条行进过程的监听
    public void onProgressChanged(SeekBar seekBar, int i, boolean b) {
        if (i == seekBar.getMax()){
            animator.pause();
        }
        if (b){//判断是否来自用户

```

```

        control.seekTo(i);
    }
}

@Override
//用户开始滑动进度条的监听
public void onStartTrackingTouch(SeekBar seekBar) {
    control.pausePlay();
    animator.pause();
}

@Override
//用户停止滑动进度条的监听
public void onStopTrackingTouch(SeekBar seekBar) {
    control.continuePlay();
    animator.resume();
}
}

```

}

in Song5.java:

```

package com.example.myapplication9;

import android.animation.ObjectAnimator;
import android.content.ComponentName;
import android.content.Intent;
import android.content.ServiceConnection;
import android.os.Bundle;
import android.os.Handler;
import android.os.IBinder;
import android.os.Looper;
import android.os.Message;
import android.view.View;
import android.view.animation.LinearInterpolator;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.SeekBar;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

public class song5 extends AppCompatActivity{

```

```

private ImageView iv_cover;
private static SeekBar sb;
private static TextView tv_progress,tv_total;
private Button btn_play,btn_pause,btn_continue,btn_exit;

private ObjectAnimator animator; //声明一个动画组件 ObjectAnimator

private MediaPlayer5.MusicControl control;//声明 MusicService 中的音乐控制器

private ServiceConnection connection = new ServiceConnection() { //声明服务连接
    @Override
    public void onServiceConnected(ComponentName componentName, IBinder iBinder) {
        control = (MediaPlayer5.MusicControl) iBinder;//实例化音乐控制对象，即 control。
    }
    @Override
    public void onServiceDisconnected(ComponentName componentName) {

    }
};

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_song5);
    init();
}

public void init(){
    iv_cover = findViewById(R.id.iv_cover5);
    sb = findViewById(R.id.sb5);
    tv_progress = findViewById(R.id.tv_progress5);
    tv_total = findViewById(R.id.tv_total5);

    btn_play = findViewById(R.id.btn_play5);
    btn_pause = findViewById(R.id.btn_pause5);
    btn_continue = findViewById(R.id.btn_continue5);
    btn_exit = findViewById(R.id.btn_exit5);

    OnClickListener monclick = new OnClickListener();
    btn_play.setOnClickListener(monclick);
    btn_pause.setOnClickListener(monclick);
    btn_continue.setOnClickListener(monclick);
    btn_exit.setOnClickListener(monclick);
}

```

//执行动画的对象是 iv_cover, // 动画效果是 0-360° 旋转 (用的是浮点数, 所以加个f)。

```
animator = ObjectAnimator.ofFloat(iv_cover,"rotation",0.0f,360.0f);
animator.setDuration(10000); //旋转一周的时长, 单位是毫秒, 此处设置了 10s
animator.setInterpolator(new LinearInterpolator()); //设置匀速转动
animator.setRepeatCount(-1); //设置循环, 此处设置的是无限循环。如果是正值, 意味着转动多少圈。
```

//声明一个意图, 该意图进行服务的启动, 意思是将 MusicService 里面的服务要传到主程序这里来。

```
Intent mintent = new Intent(song5.this,MusicPlayer5.class);
bindService(mintent,connection,BIND_AUTO_CREATE); //建立意图中 MainActivity 与 MusicService 两对象的服务连接
```

```
seekBarListener msbListener = new seekBarListener();
sb.setOnSeekBarChangeListener(msbListener);
```

```
}
```

// 设置播放、暂停、继续和退出按钮的监听 (或点击) 事件

```
class OnClick implements View.OnClickListener{
```

```
@Override
```

```
public void onClick(View view) {
    switch (view.getId()){
        case R.id.btn_play5:
            //播放音乐
            control.play();
            //光盘开始转
            animator.start();
            break;
        case R.id.btn_pause5:
            //停止播放音乐
            control.pausePlay();
            //光盘停止转
            animator.pause();
            break;
        case R.id.btn_continue5:
            //继续播放音乐
            control.continuePlay();
            //光盘继续转
            animator.resume();
            break;
        case R.id.btn_exit5:
            finish();
            break;
```

```

    }
}
}

```

```

@Override
protected void onDestroy() {
    control.stopPlay();
    unbindService(connection);
    super.onDestroy();
}

```

//Handler 主要用于异步消息的处理，在这里是处理子线程 MusicService 传来的消息

```

public static Handler handler = new Handler(Looper.getMainLooper()){

```

```

    @Override
    public void handleMessage(@NonNull Message msg) {
        //super.handleMessage(msg);
        Bundle bundle = msg.getData();
        int duration = bundle.getInt("duration");//把音乐时长放在 bundle 里
        int currentDuration = bundle.getInt("currentDuration");//把音乐当前播放时长放在
        bundle 里

```

```

        sb.setMax(duration);
        sb.setProgress(currentDuration);

```

//显示总时长

```

        int minite = duration / 1000 / 60;
        int second = duration / 1000 % 60;
        String strMinite = "";
        String strSecond = "";
        if (minite < 10){
            strMinite = "0" + minite;
        }else {
            strMinite = minite + "";
        }
        if (second < 10){
            strSecond = "0" + second;
        }else {
            strSecond = second + "";
        }
        tv_total.setText(strMinite + ":" + strSecond);

```

```

//显示播放时长
minite = currentDuration / 1000 /60;
second = currentDuration / 1000 % 60;

if (minite < 10){
    strMinite = "0" +minite;
}else {
    strMinite = minite + "";
}
if (second < 10){
    strSecond = "0" + second;
}else {
    strSecond = second + "";
}
tv_progress.setText(strMinite + ":" + strSecond);
}
};

//给进度条设置监听
class seekBarListener implements SeekBar.OnSeekBarChangeListener {
    @Override
    //进度条行进过程的监听
    public void onProgressChanged(SeekBar seekBar, int i, boolean b) {
        if (i == seekBar.getMax()){
            animator.pause();
        }
        if (b){//判断是否来自用户
            control.seekTo(i);
        }
    }

    @Override
    //用户开始滑动进度条的监听
    public void onStartTrackingTouch(SeekBar seekBar) {
        control.pausePlay();
        animator.pause();
    }

    @Override
    //用户停止滑动进度条的监听
    public void onStopTrackingTouch(SeekBar seekBar) {
        control.continuePlay();
        animator.resume();
    }
}

```


}

}