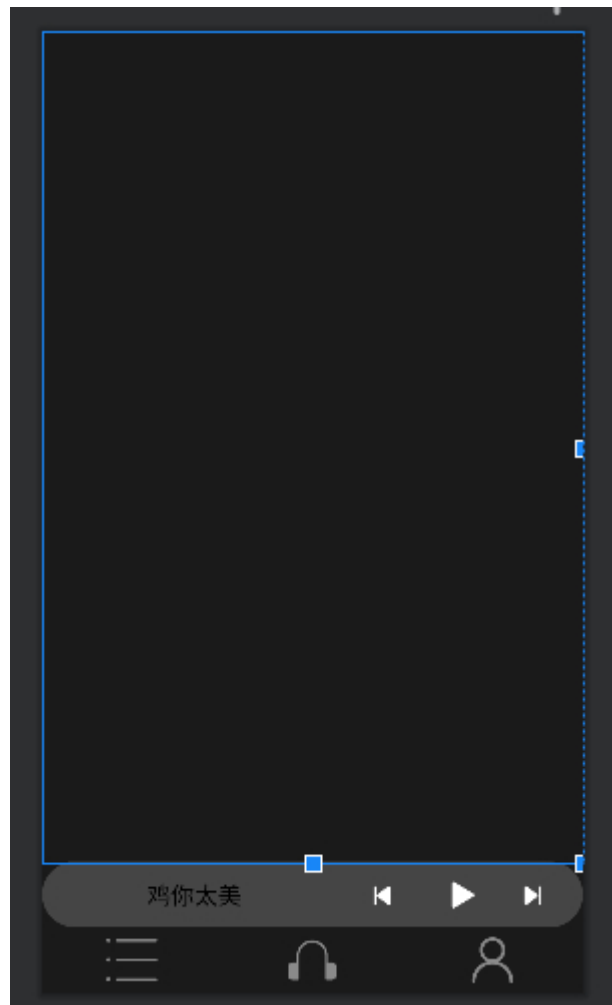


# activity\_main

从主页面出发，一整个装载的是一个ViewPager

```
1 activity_main
2 <androidx.viewpager.widget.ViewPager
3     android:background="#1A1A1A"
4     android:id="@+id/vp"
5     android:layout_width="match_parent"
6     android:layout_height="0dp"
7     android:layout_weight="1" />
```



ViewPager内部载入三个fragment.xml页面(music音乐列表界面 find视频列表界面 user个人信息界面)

在xml页面当中仅有一个RecyclerView

## fragment\_music.java

以fragment\_music.java举例，首先调用 `onViewCreated` 方法显示页面

在再创建广播监听管理，监听整个页面的单击操作(`mItemViewListClickReceiver`)与资源刷新操作(`intentFilter`)

创建广播接收器**BroadcastReceiver mItemViewListClickReceiver**的线程用于接收用户数据当中的喜欢列表，并将数据传输到主线程当中，最后初始化歌曲列表。

```
broadcastManager.registerReceiver(mItemViewListClickReceiver, intentFilter);
```

## 重写MainActivity的onResume方法

从其他页面返回到歌曲MainActivity刷新一遍喜欢的歌曲，且将听过的歌保存数据到本地实现历史播放

## initdata方法

获取全部的音乐资源，使用handler发送Message到主线程

onCreate与onCreateView是继承Fragment类必须重写的两个方法 不做讲解

handler操作

- 1.获取音乐数据并存放进application.data当中
- 2.将recyclerView的布局设置一个布局管理器，管理器装配的是线性布局管理器

```
1  LinearLayoutManager layoutManager = new LinearLayoutManager(getActivity());
2      recyclerView.setLayoutManager(layoutManager);
```

3.创建适配器

```
1  adapter = new music_Adapter(data);
```

在适配器当中

```
1  //指定一个条目的布局文件来创建ViewHolder,ViewHolder相当于一个容器，它持有一个条目中所拥有的的组件
2  public music_Adapter.InnerHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType)
3  {
4      View view = View.inflate(parent.getContext(), R.layout.item_music,null);
5      return new InnerHolder(view);
6  }
```

item\_music条目布局内容自己定义，其中的组件自己做调整

```
1  //为每个条目布局绑定数据
2  public void onBindViewHolder(@NonNull music_Adapter.InnerHolder holder, int position) {
3      holder.setData(mdata.get(position),position);
4  }
```

数据即为当前被点击条目的数据，传入的是歌曲列表中，当前被点击的条目的位置作为下标，获取该下标处的所有信息。

```

1 //返回条目个数
2 public int getItemCount() {
3     if (mdata != null){
4         return mdata.size();
5     }
6     return 0;
7 }

```

```

1 //自定义的ViewHolder，它持有条目布局文件中的组件
2 public class InnerHolder extends RecyclerView.ViewHolder{
3     private int mposition;
4     private String murl;
5     private String mname;
6     private String mgeshou;
7     private String mmurl;
8     private int mvposition=9999;
9     private String mv_number;
10
11 //构造方法
12 public InnerHolder(@NonNull View itemView) {
13     super(itemView);
14     //绑定控件ID
15     mimageView=itemView.findViewById(R.id.item_image_music);
16     text_ge = itemView.findViewById(R.id.item_text_music_name);
17     text_name = itemView.findViewById(R.id.item_text_name);
18     love = itemView.findViewById(R.id.love);
19     mv = itemView.findViewById(R.id.mv);
20     //mv按键单击事件
21     mv.setOnClickListener(new View.OnClickListener() {
22         @Override
23         public void onClick(View view) {
24             if (mvposition!=9999)
25             {
26                 mvclick.mvclick(mname,mdata.get(mposition).mv_url,mv_number);
27             }else{
28                 if (monIteamClickListener != null) {
29
30                     monIteamClickListener.onIteamClick(mposition,murl,mname,mgeshou,mmurl);
31                 }
32             }
33         }
34     });
35     app = application.getInstance();
36     //喜欢单击事件
37     love.setOnClickListener(new View.OnClickListener() {
38         @Override
39         public void onClick(View view) {
40             mclick.inClick(mdata.get(mposition),itemView);
41         }
42     });
43     //条目单击事件
44     itemView.setOnClickListener(new View.OnClickListener() {
45         @Override
46         public void onClick(View view) {
47             if (monIteamClickListener != null) {
48
49                 monIteamClickListener.onIteamClick(mposition,murl,mname,mgeshou,mmurl);

```

```
49         }  
50     }  
51     });  
52 }
```

## 主要思路

ViewPager内 装配了三个Fragment碎片 (fragment\_music.java fragment\_find.java fragment\_user.java), 其中音乐碎片与视频碎片内 各装配了一个适配器 (video\_Adapter.java music\_Adapter.java), 适配器内装配的是相关条目, 视频适配器装配的是视频页条目布局(item\_video.xml), 音乐适配器装配的是音乐页条目布局(item\_video.xml), 且在Fragment碎片当中设置条目或者按钮被单击的事件, 条目被点击之后, 传入当前条目的位置, 根据位置当做下标获取当前数据列表当中的数据。使用Intent(**组件间互相通信的信息对象**)显式传输到对应的播放器Activity(music\_player video\_player) 播放器获取到歌曲信息播放对应歌曲资源, 使用自带的音视频播放器组件来控制资源的播放控制 视频则是调用外部库的插件来进行播放控制

### 登录注册阶段

登录即先通过登录的逻辑判断 如是否勾选 手机号位数判断 密码是否为空 经过逻辑判断之后则使用mysql服务类来检索数据库中的信息, 如果有对应信息, 则返回成功 修改Message事件码 做出登录成功的响应  
注册即先通过注册的逻辑判断 如判断手机号位数是否正确 密码是否为空 (修改事件码0) -> 判断手机号是否存在(修改事件码1) -> 用户名是否存在(修改事件码2) -> 注册用户 -> 做出注册与否的响应