

Android 音乐APP核心功能的实现（扫描本地、Service播放、循环模式、进度更新、列表增删等）



眼看心勿动 (/u/c110c6dcd223) 作者 + 关注

2016.09.14 18:25 字数 2450 阅读 1058 评论 3 喜欢 23

(/u/c110c6dcd223)

一、前言

自己做的音乐APP的界面部分已完成，各种功能基本实现了。但是我对音乐后台播放这一块（Service）不太满意。因为之前对Service接触得不多，所以没有一个明确的规划，想要实现什么就往Service里添加新的需要实现的功能，导致原来的Service的代码十分臃肿和凌乱。

一款音乐APP，最核心的业务就是播放音乐，所以设计好一个完善的音乐APP的Service十分有必要。自己在动手写的时候参考了网上许多的例子，Github里完整的项目，由于编码风格和自己的不太一样，理解起来效率很低，一些博客里发布的仅有单个功能的源码又太简单。因此我整理了下自己的思路，做一次总结，以获得更好的提升。

二、实现功能

这个MusicService主要实现了如下功能：

基本功能

播放本地歌曲、网络歌曲，实现播放、暂停、下一首、选择循环模式、拖动进度条改变播放位置等基本功能。



读取SQLite数据

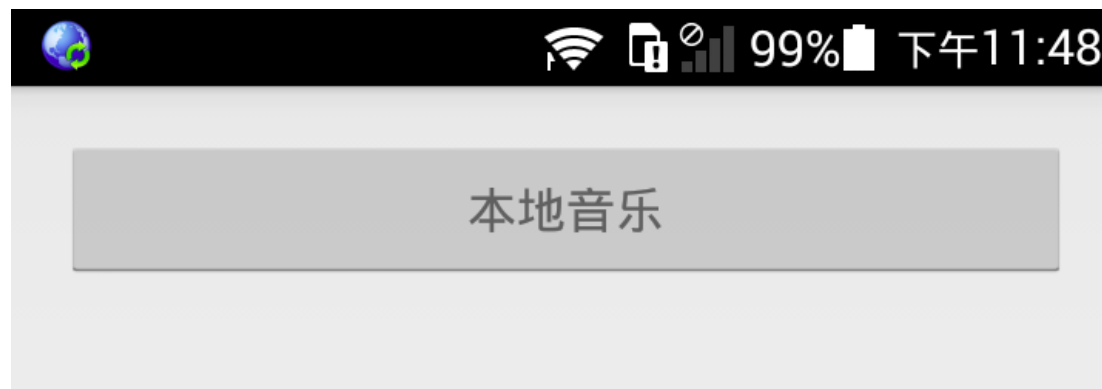
- 1、启动应用后，能够获取前一次未播放完的音乐信息（如当前时间、歌曲总长、进度条位置、歌曲、歌手、循环模式等），并装入下端播放栏中，点“播放”在暂停处续播；
- 2、进入本地音乐界面时，如之前已经启动过引用并扫描过本地音乐，则直接载入歌曲信息，如还没有扫描可点击界面的右上角弹出Dialog选择“扫描本地音乐”选项进行扫描；
- 3、在MusicService通过SQLite获得歌曲信息。

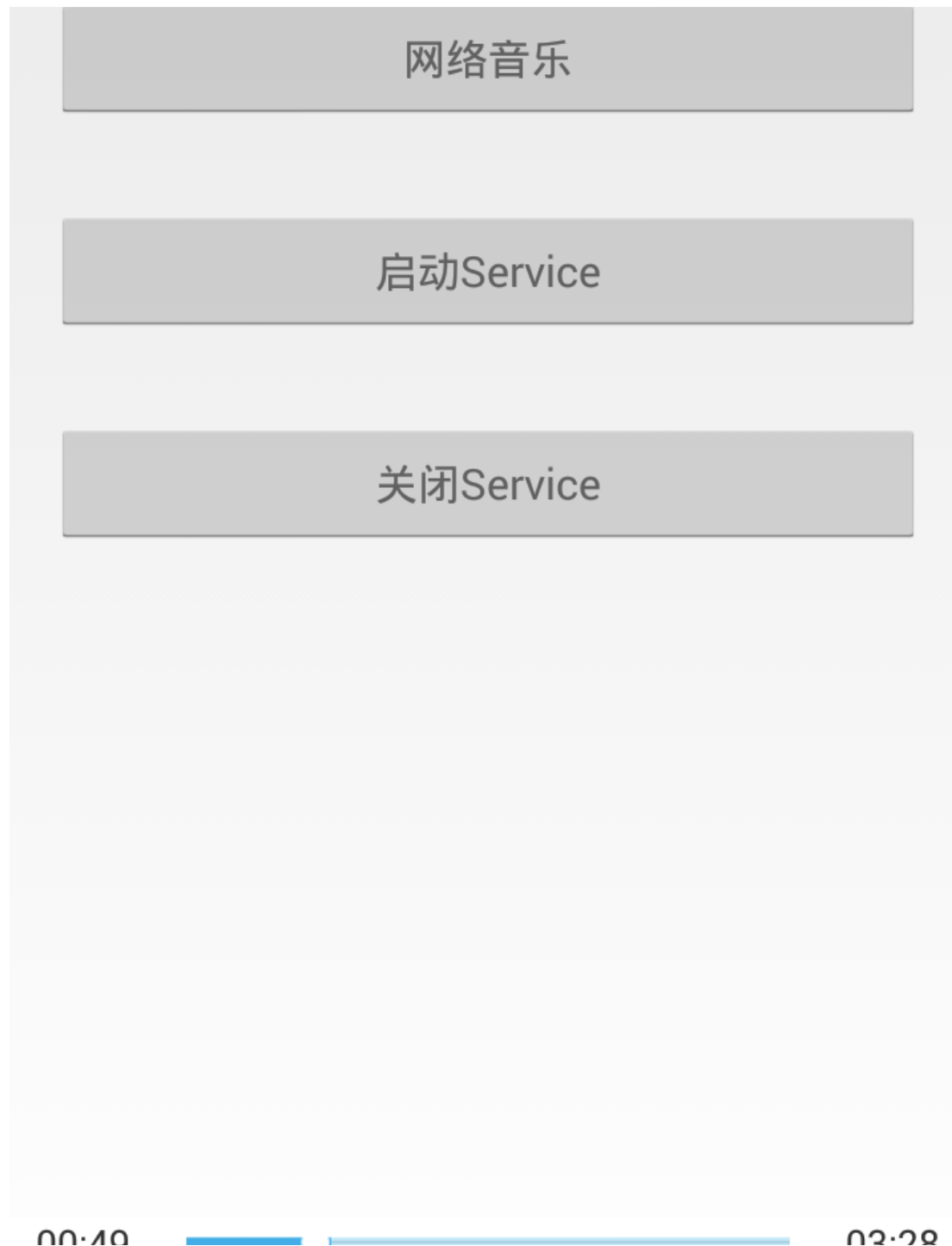
用户体验

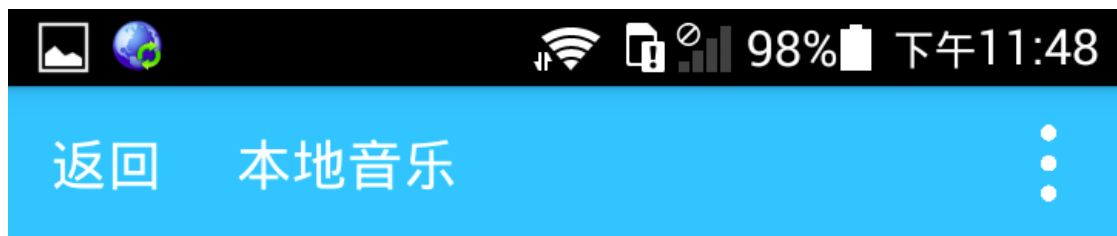
- 1、点击播放栏“列表”图标，通过DrawerLayout的方式弹出播放列表；
- 2、点击本地音乐列表或者网络歌单的歌曲即可播放，并且将当前播放的音乐自动添加至播放列表，当用户改变循环模式，或者增删本地列表、播放列表的歌曲时，MusicService均可正确地播放下一首歌曲；
- 3、通过广播发送通知，让Activity更新UI。


三、效果图

效果图如下所示：

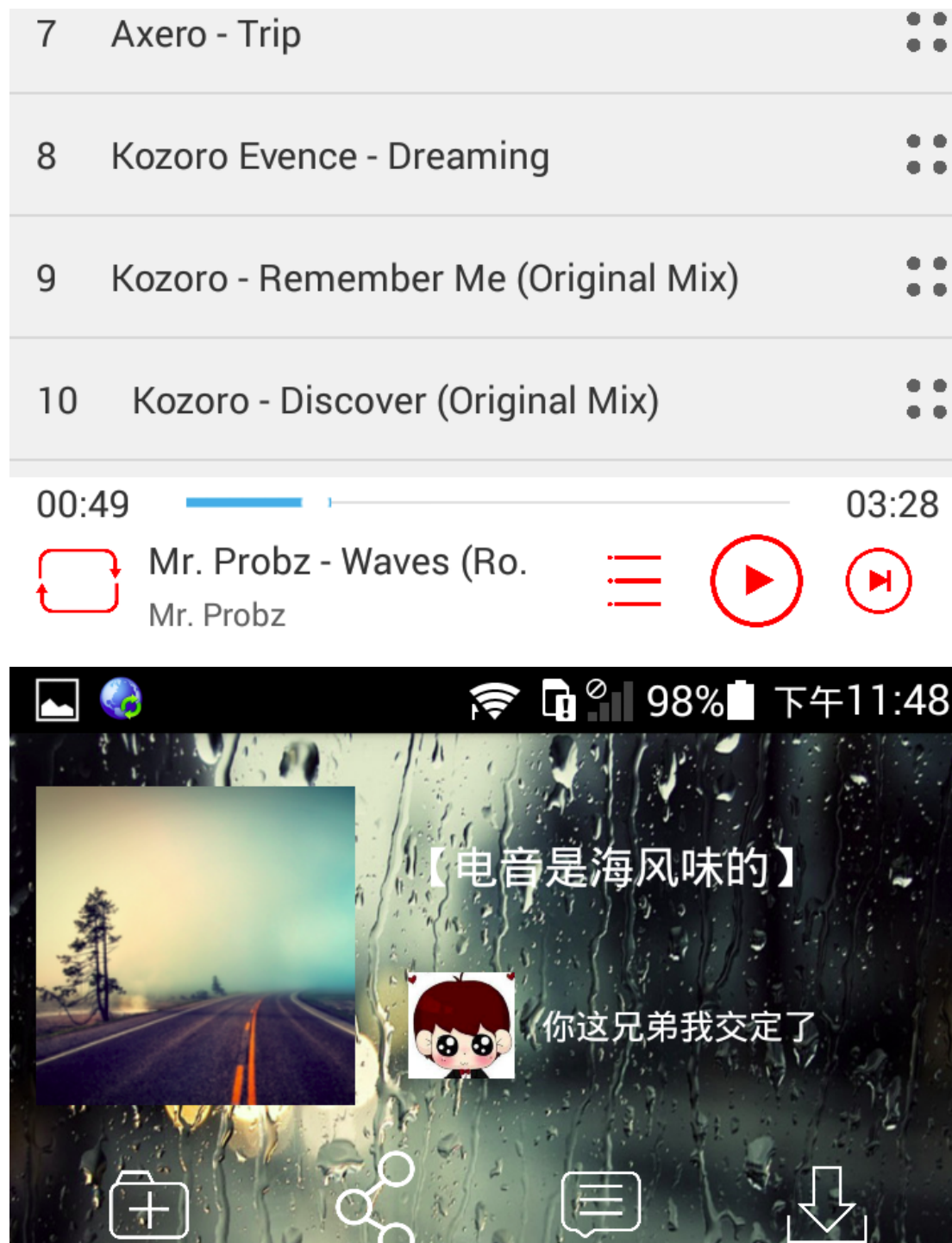




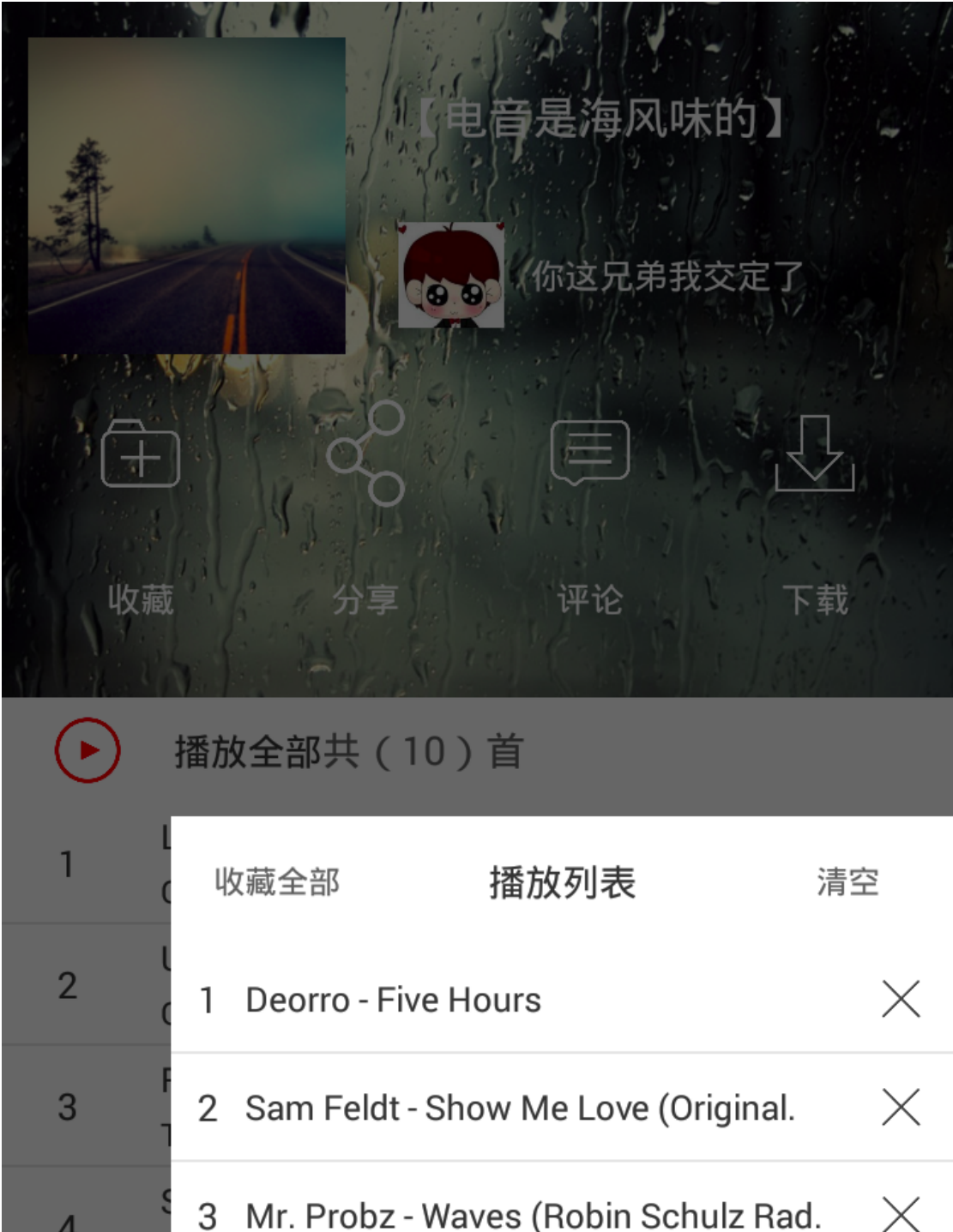


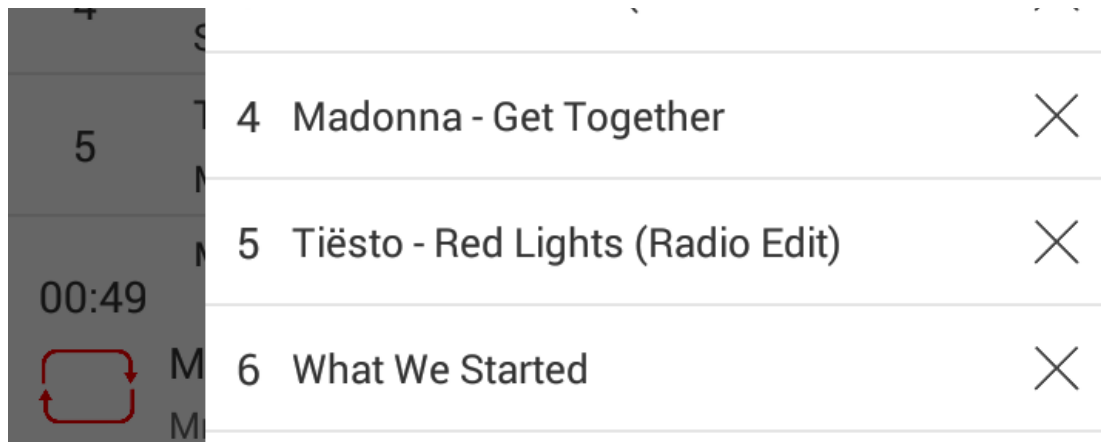
 [播放全部](#) (共24首)

1	Do_You_Wanna_Go	•• ••
2	Emo_Mamma	•• ••
3	Hit_Da_Stylin	•• ••
4	Deorro - Five Hours	•• ••
5	Deadmau5 - Pets	•• ••
6	Kozoro Skyvoice - For Us (Original Mix)	•• ••









4、关键思路

本Demo的Java文件包括：

后台Service:

MusicService，用于在后台播放音乐以及各种操作的服务类

前台Activity

Activity:MainActivity：用于进入本地音乐界面和网络音乐界面的活动类

LocalMusicActivity：显示本地音乐列表界面的活动类

OnlineMusicActivity：实现网络歌曲（也就是我APP中歌单）界面的活动类

还有一些工具类、线程类分别用于扫描本地音乐，适配ListView等延时操作，详见源码。

此外，还需要用到SQLite去对数据进行存储和提取，我建立了一个名为music.db的数据库，并在这个数据库里建立了playerbarinfo、localmusic、onlinemusic、playlist、loadlocalmusiclistview这几个表格，分别管理播放栏数据、本地音乐、网络

音乐、播放列表音乐、是否曾扫描并载入本地音乐等信息。

还需要在Activity里定义localsong_list、onlinesong_list和play_list分别存储本地音乐、网络歌曲以及播放列表的歌曲信息。

代码运行流程就是：

·初始化

启动Activity时先对数据库里的playerbarinfotb这一个表格中去搜索，如果存在数据，就将播放栏各项信息提取出来，初始化播放栏。

·进入本地

进入本地音乐后，搜索loadlocalmusiclistviewtb，如果存在数据，说明已经扫描过本地音乐，因此从localmusicdb里将全部的歌曲信息提取出来，载入本地音乐界面的ListView中，如果loadlocalmusiclistviewtb不存在数据，就显示“扫描本地”按钮，对手机上的本地音乐进行扫描，并在扫描完成后将扫描结果载入ListView。

在MusicService里，用public static final String的语句定义好各种动作，如：

```
public static final String ACTION_UPDATE_TIME="ACTION_UPDATE_TIME";
```

```
public static final String ACTION_PLAY_CURR_MUSIC="ACTION_PLAY_CURR_MUSIC";
```

```
public static final String ACTION_PLAY_ONLINE_MUSIC="ACTION_PLAY_ONLINE_MUSIC";
```

```
public static final String ACTION_PLAY_LOCAL_MUSIC="ACTION_PLAY_LOCAL_MUSIC";
```



```
public static final String ACTION_PLAY_PLAYLIST_MUSIC="ACTION_PLAY_PLAYLIST_MUSIC";
```

```
public static final String ACTION_DELETE_LOCALMUSIC="ACTION_DELET_LOCALMUSIC";
```

```
public static final String ACTION_DELETE_PLAYLIST_MUSIC="ACTION_DELETE_PLAYLIST_MUSIC";
```

```
public static final String ACTION_PLAY_NEXT="ACTION_PLAY_NEXT";
```

```
public static final String ACTION_PLAY_ALL_LOCALMUSIC="ACTION_PLAY_ALL_LOCALMUSIC";
```

```
public static final String ACTION_PLAY_ALL_ONLINEMUSIC="ACTION_PLAY_ALL_ONLINEMUSIC";
```

```
public static final String ACTION_CLEAR_ALL_PLAYLIST="ACTION_CLEAR_ALL_PLAYLIST";
```

```
public static final String ACTION_CHANGE_CIRCULATE_MODE="ACTION_CHANGE_CIRCULATE_MODE";
```

这些动作顾名思义，分别是“更新时间”、“播放当前播放栏音乐”、“播放网络音乐”、“播放本地音乐”、“播放音乐列表里的音乐”、“删除本地歌曲”、“删除播放列表里的歌曲”、“播放下一首”、“播放所有本地音乐”、“播放所有网络音乐”、“清空播放列表”等动作。



·播放歌曲

如果点击了本地音乐列表里的某一首歌，就通过

```
intent.setAction(MusicService.ACTION_PLAY_LOCAL_MUSIC);
```

的方法，向intent添加了代表了“播放本地音乐”的这个动作标示，与此同时还通过

```
Intetn.putExtra("position",position);
```

的方法传入当前点击的歌曲在本地歌曲列表中的位置，接着用

```
startService(intent);
```

的方法启动项目的服务类MusicService。MusicService在onStartCommand（）方法里通过

```
if(intent.setAction.equals(MusicService.ACTION_PLAY_LOCAL_MUSIC);
```

的语句去判断传入的动作标识是什么，并执行相应的操作。例如播放列表里有歌曲

A、B、C、D、E顺序排列，他们的播放路径分别

是“sdcard/a.mp3”、“sdcard/b.mp3”、“sdcard/c.mp3”、“sdcard/d.mp3”、“sdcard/e.mp3”

，用户点击的是本地音乐列表里的第二首歌，MusicService就开始获取传入的值

position，为1（从0开始算起），在MusicService里提取本地音乐的数据库，通过

position找到用户点击的是歌曲B。然后播放这首歌。

·添加

歌曲播放的同时，程序会对播放列表的数据库playlistb进行搜索，如果playlistb里面没有当前播放的歌曲，那么用insert语句将其添加至playlistb，用.add()语句添加至play_list，并通过Adapter的.updateData()方法刷新播放列表的ListView界面。

歌曲播放时通过定时线程，每秒发送一次播放器信息广播，当前Activity接收到广播后更新播放栏的信息。



·下一首

歌曲开始播放之后，在MusicService里执行getNextSong()的方法，根据循环模式，获取下一首播放的歌曲。列表循环的时候，获取歌曲B的下一首也即歌曲C，单曲循环的时候，获取夏一首歌曲也即歌曲B，随机播放的时候，通过Random取出一个范围为0到(play_list.size()-1)的随机数m，用play_list.get(m)的方法获取下一首待播歌曲。当监听到当前歌曲播放完毕后，执行播放下一首。

·删除

当删除本地音乐或者播放列表里的歌曲C时，也获取音乐的position，为2，并发送删除歌曲的动作标示到MusicService，MusicService根据position判断删除的是歌曲C，如果当前循环模式是列表循环，删中的正好是下一首待播的歌曲，那么就通过歌曲的路径，获取另外的正确的下一首待播歌曲，得到歌曲D（接着D又从播放列表中被删除，getNextSong()又获取歌曲E）

·循环

在MusicService里定义一个公共的int变量MUSIC_CIRCULATION_MODE，设置终态变量来表示循环模式，如public static final int LIST_CIRCULATION=0表示列表循环，public static final int SINGLE_CIRCULATION=1表示单曲循环，public static final int SHAFFULE=2表示随机播放。点击播放栏的循环图标可以改变MusicService的公共变量MUSIC_CIRCULATION_MODE，点击一次就轮次赋值列表循环、单曲循环、随机播放这三个常量，并通过动作标示启动MusicService，MusicService根据switch(MUSIC_CIRCULATION_MODE)，启动getNextSong()方法去获取下一首待播歌曲。

·网络歌单

网络歌单的页面在我自己的APP里是通过一个HttpThread的线程类去获取我服务端上的歌单列表的，在这个源码里我就直接定义一个list并直接往里面添加歌曲信息了。只是歌曲的URL是我家局域网里服务端的歌曲地址，如有需要可以直接改成别的网络歌曲连接。



·MainActivity

在这个源码中我并没有设置从本地音乐界面或网络歌单界面跳转回主界面的方法，返回主界面上通过菜单上的返回键进行的，因此，为解决本地音乐界面或者网络歌单界面返回主界面后，主界面由于没有重新onCreate()而没有更新播放列表的问题，我重写了onResume()方法，加入了一个刷新播放列表的ListView的方法。使得几个界面的播放列表能保持同步。

5、源码

安卓音乐APP后台Service源码 (<https://github.com/wmy0601034/MusicService>)

📖 日记本 (/nb/6269060)

举报文章 © 著作权归作者所有



眼看心勿动 (/u/c110c6dcd223)

写了 5018 字，被 7 人关注，获得了 23 个喜欢

(/u/c110c6dcd223)

+ 关注

♡ 喜欢 (/sign_in?utm_source=desktop&utm_medium=not-signed-in-like-button) | 23



更多分享

(http://cwb.assets.jianshu.io/notes/images/5795362/weibo/image_2)

被以下专题收入，发现更多相似内容



Android... (/c/63c8c0a9de71?utm_source=desktop&utm_medium=notes-included-collection)



