文本复制检测报告单(全文标明引文)

№:ADBD2018R_20180530153127201805301548194401738907 25　　检测时间:2018-05-30 15:48:19

检测文献:　53140122_杨峰_计算机科学与技术_音乐播放器的微信小程序的设计与实现

作者:　杨峰

检测范围:　中国学术期刊网络出版总库

中国博士学位论文全文数据库/中国优秀硕士学位论文全文数据库

中国重要会议论文全文数据库

中国重要报纸全文数据库

中国专利全文数据库

图书资源

优先出版文献库

大学生论文联合比对库

互联网资源(包含贴吧等论坛资源)

英文数据库(涵盖期刊、博硕、会议的英文数据以及德国Springer、英国Taylor&Francis 期刊数据库等)

港澳台学术文献库

互联网文档资源

CNKI大成编客-原创作品库

个人比对库

时间范围:　1900-01-01至2018-05-30

检测结果

总文字复制比：0.4%　　跨语言检测结果：0%

去除引用文献复制比：0.4%　　去除本人已发表文献复制比：0.4%

单篇最大文字复制比：0.3%

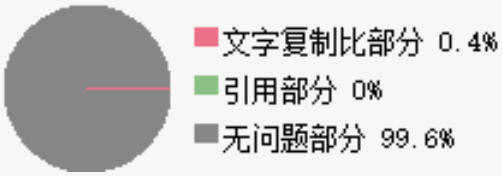
重复字数：[126]　　总段落数：[7]

总字数：[30175]　　疑似段落数：[1]

单篇最大重复字数：[90]　　前部重合字数：[28]

疑似段落最大重合字数：[126]　　后部重合字数：[98]

疑似段落最小重合字数：[126]



指　标：疑似剽窃观点　疑似剽窃文字表述　疑似自我剽窃　疑似整体剽窃　过度引用

表格：0　　公式：0　　疑似文字的图片：0　　脚注与尾注：0

0%（0）　53140122_杨峰_计算机科学与技术_音乐播放器的微信小程序的设计与实现_第1部分（总2117字）

0%（0）　53140122_杨峰_计算机科学与技术_音乐播放器的微信小程序的设计与实现_第2部分（总2564字）

1.6%（126）　53140122_杨峰_计算机科学与技术_音乐播放器的微信小程序的设计与实现_第3部分（总7854字）

0%（0）　53140122_杨峰_计算机科学与技术_音乐播放器的微信小程序的设计与实现_第4部分（总9984字）

0%（0）　53140122_杨峰_计算机科学与技术_音乐播放器的微信小程序的设计与实现_第5部分（总6048字）

0%（0）　53140122_杨峰_计算机科学与技术_音乐播放器的微信小程序的设计与实现_第6部分（总742字）

0%（0）　53140122_杨峰_计算机科学与技术_音乐播放器的微信小程序的设计与实现_第7部分（总866字）

（注释：无问题部分　文字复制比部分　引用部分）

1. 53140122_杨峰_计算机科学与技术_音乐播放器的微信小程序的设计与实现_第1部分　总字数：2117

相似文献列表　文字复制比：0%(0)　疑似剽窃观点：(0)

原文内容　红色文字表示存在文字复制现象的内容;绿色文字表示其中标明了引用的内容

吉林大学学士学位论文（设计）承诺书

本人郑重承诺：所呈交的学士学位毕业论文（设计），是本人在指导教师的指导下，独立进行实验、设计、调研等工作基础上取得的成果。除文中已经注明引用的内容外，本论文（设计）不包含任何其他个人或集体已经发表或撰写的作品成果。

对本人实验或设计中做出重要贡献的个人或集体，均已在文中以明确的方式注明。本人完全意识到本承诺书的法律结果由本人承担。

学士学位论文（设计）作者签名：杨峰

2018年5月20日

摘要

音乐播放器的微信小程序的设计与实现

随着物质生活的提高，人们在精神层面上也有了更多的要求，在闲暇的时光里，听音乐无疑是最直接的陶冶情操，放松身心的一种方式。正是这一需求的不断提高，也就极大地促进了软件媒体的发展，各种音乐播放器软件如雨后春笋般的出现。本毕业设计是以微信小程序开发平台为基础，利用微信小程序官方提供的开发者工具编写的一款在线音乐播放器，包括研究的背景与意义，现状分析以及发展趋势，对微信小程序架构的介绍，音乐播放器小程序的设计与开发以及最后的功能测试这几大模块。这款音乐播放器的功能设计旨在简约实用，包括新歌推荐，流行音乐排行榜，歌曲搜索这三大核心功能，既能满足用户日常的需求，也不会显得过于臃肿复杂，让用户在使用微信的过程中即可随时随地享受音乐带给生活的惬意。

关键字：微信小程序，音乐播放器，在线音乐

Abstract

Design and Implementation of a Music Player Based on Wechat Applet

With the improvement of material life, people have more requirements on the spiritual level. In leisure time, listening to music is undoubtedly the most direct way to cultivate sentiment and relax. It is this constant increase in demand that has greatly promoted the development of software media. Various music player software has mushroomed. The graduation design is based on the WeChat applet development platform, an online music player written using the developer tools provided by the WeChat applet, including the research background and significance, status analysis and development trend, the introduction of the WeChat applet architecture., the design and development of music player applets and the final functional testing of these major modules. The function of this music player is designed to be simple and practical, including new song recommendations, pop music charts, song search, these three core functions, both to meet the daily needs of users, it will not appear too bloated and complex, allowing users to use In the process of WeChat, you can enjoy the music at any time and anywhere.

Keywords: Wechat, Music Player, Online Music

目录

2. 53140122_杨峰_计算机科学与技术_音乐播放器的微信小程序的设计与实现_第2部分 总字数：2564

相似文献列表 文字复制比：0%(0) 疑似剽窃观点：(0)

原文内容 红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

第1章绪论

1.1背景与意义

在现在这个生活节奏变得越来越快的时代里，我们的科技水平也跟着发展得越来越高端，随着近几年人工智能的大力发展，全球智能化是未来世界的必然趋势，在这一过程当中，手机则是现代人们普遍最享受智能化的一件产物。随着人们不再满足于只能用手机来进行打电话，发短信等通讯活动，以及移动互联网的飞速发展，智能化手机逐渐普遍了起来，各种各样的应用也开始占据我们的手机，聊天影音购物等各种满足人们需求的app应运而生，而这些手机软件的普及也表现出传统互联网正在逐渐朝着移动互联网转型，人们越来越习惯于使用手机来代替电脑。随着Android应用开发的大火，各种app充斥着我们的手机，而当现在我们谈到手机上的常用的app 时，微信绝对是不可忽视的存在。在张小龙团队于2011年上线微信之后，这款在智能手机上可以得到即时通讯服务的社交软件，在短暂的的时间里迅速吸收用户，而且不仅是年轻人使用微信，中老年人也成为了微信的受众群体，微信几乎已经成为每个智能手机必不可少的应用。很多人的关系链已经从手机通讯录转移到了微信，人们用微信发语音聊天、刷朋友圈，走路在用微信，乘车在用微信，甚至吃饭娱乐都会拍照上传到微信朋友圈，微信已经成为我们生活中密切相关的移动应用。而在2016年微信所推出的小程序功能， 因为拥有“触手可及，用完即走”这样的特性而被人们广泛关注，用户可以通过搜索或者扫描二维码即可找到并使用相应的应用功能，正是因为有了这一特性，广大用户可以不用担心手机上太多应用占据空间的问题了。相比微信公众号，微信小程序所提供的不仅有更加丰富的框架组件，还有大量的api接口提供给开发者来使用，包括导航、重力感应、位置以及网络等。通过使用这些已经供给好的的组件和接口，开发者们能够更加便捷的编写自己的应用。至此微信小程序开始展现它不一样的吸引力。

而在这个每个人都忙着努力奋斗的时代，快速的生活节奏也在不断地压榨着我们的精力，在片刻的闲暇时光里，人们越来越习惯于碎片式的休闲方式。在手机上看一些即时性的新闻，刷一刷微博，打一两把手游等，都是人们常见的放松方法，而最常见的则不得不提到聆听音乐了。音乐往往是与我们的情绪紧密的联系在一起的，在感到疲惫时，我们会播放一些平静悠扬的轻音乐抚慰自己的心灵，在感到愉悦时，我们会习惯于播放一些欢快的流行音乐，随着音乐的节奏一起哼上几句。音乐使阴郁的情绪变得明朗，躁动的情绪变得安宁，很难想象我们的生活中缺少了音乐会是什么样子。而随着市面上各式各样的音乐播放器不断涌出，怎样去选择一款适合自己的音乐播放器也是一件需要反复斟酌的事情。现在许多的音乐播放器产品，一味地追求功能的多样化，样式的新奇美观，反而造成其自身所占空间过大，许多的功能用户并不需要用到，因而显得繁杂臃肿。在这样的环境背景下，使用微信小程序来开发一款简约的音乐播放器，既可以缓解手机内存的压力，又不至于让用户对过多的功能感到厌烦。 同时对于越来越多中老年人的人们也都开始使用微信的背景下，不需要那么多复杂的操作就可以在使用微信的过程中享受音乐，具有很强的实用性。

1.2现状分析和发展趋势

微信以令人咂舌的速度从一款最初的社交媒体发展为广大用户的日常生活中必不可少的通信联络工具和获取信息的平台，目前的微信小程序基于9亿月活跃用户的微信生态中，入口多，流量大而且功能简单便捷，围绕着小程序展开的生态工具的建设以及开发将有可能成为未来移动互联网的一个相当大的机遇。在微信小程序的发布之初，张小龙团队曾提出过“好的产品应该是用完即走的”“让商业存在于无形之中”这两个理念。关于“好的产品应该是用完即走的”这个理念，张小龙认为一个好的产品绝不是要一直粘住用户，而是在用户需要的时候及时出现，使用完毕之后自动消失不会在用户生活中产生什么影响，对于用户来说这个模式是最容易被接受的。“一个好的产品的商业化和用户的价值以及好的体验是不矛盾的，好的商业化应该是不骚扰用户，并且是只触达他想要触达的那一部分用户”，张小龙在阐述“让商业化存在于无形之中”的价值观时提到了这句话。正是因为有着这样的价值观和开发理念，现在我们所看到的小程序拥有着“触手可及”以及“用完即走”这两大核心特性。用户只需要通过扫描二维码或在微信中搜索，不用安装即可使用小程序，使用完毕就退出，整个过程中完全不用担心手机内存的问题，让用户的日常生活变得更为便捷。

据最新的数据分析统计，微信小程序每个月的活跃用户数已经达到了4亿，而且每天的日活跃用户数稳定在1.4亿左右

，甚至有可能在短期时间内突破至2亿，这些数据足以说明用户对小程序的认可。从2017年年底风靡一时的“跳一跳”小程序，到现如今微信里随处可见的各种小游戏小程序分享，这一井喷式的发展自然少不了微信对于自家小程序的大力支持。目前在小程序中有多达50个入口，这使其充分的完成了线上和线下的场景连接，正是这一重大突破让小程序逐渐成为人们在娱乐休闲以及购物等各个方面的不可或缺的重要工具。在这个移动互联网时代，微信确实是一个不可或缺的应用，而微信小程序的出现恰好切合了当前时代的需要，毋庸置疑将会成为政府、企业机构以及开发们争相开发使用的互联网应用场景。

在音乐播放器这一方面，当前的应用市场上充斥着各种流行的手机音乐产品，包括老牌的QQ音乐和多米音乐，以及近几年大火的网易云和虾米音乐等，有竞争自然就有了不断发展创新的动力。而随着泛娱乐这个概念的提出，网络文学，直播行业以及越来越多年轻人接受的二次元文化，游戏等的大热，人们对各种类型的音频文件接受程度的的提升，让音乐播放器的涉及领域不断的扩大，如今的音乐播放器早已不是单纯的播放音乐了。现如今的在线音乐这个领域，有着国内外不断增强的版权意识，有着全球化完备的新兴市场，特别还有着国家的大力支持，不难看出现在甚至在未来很长的一段时间，都有着很长足的发展空间。而且随着无线通信技术和Wi-Fi的飞速发展，随时随地掏出手机就能连上wifi或者4G，越来越多的用户开始习惯于在线收听音乐而不需要下载到本地进行收听，而且这一行为在中国移动，联通，电信三大运营商相继推出专属应用免流量的日租卡之后则更为普遍，可以看出轻便实用是未来音乐播放器的一大发展趋势。

3. 53140122_杨峰_计算机科学与技术_音乐播放器的微信小程序的设计与实现_第3部分　总字数：7854

相似文献列表　文字复制比：1.6%(126)　疑似剽窃观点：(0)

1	基于微信小程序的在线阅读平台的开发 曹西梅 - 《大学生论文联合比对库》- 2017-04-11	1.1%（90） 是否引证：否
2	第一讲 Java程序设计概述-百度文库 -《互联网文档资源（http://wenku.baidu.c）》- 2012	0.4%（34） 是否引证：否
3	P130510794_盛琬婷_2013j级软件工程2班_基于微信小程序的在线阅读平台开发 盛琬婷 - 《大学生论文联合比对库》- 2017-04-17	0.4%（29） 是否引证：否

原文内容　红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

第2章微信小程序架构分析

2.1微信小程序框架结构

一个完整的软件系统的框架结构应该是多层次的，一般说来分为数据访问层，业务逻辑层和表示层。从本质上来说，微信小程序属于软件系统中的表示层，相当于前端页面，所以一般还需要有数据访问层和业务逻辑层这些后台系统为微信小程序提供服务。而随着前端程序开发的进一步发展，近几年前端程序开发也有了层次的细分，微信小程序就是这样的前端开发框架，提供了前端的逻辑层、视图层的分层结构。

(1)目录结构

在一个网页的开发过程中，需要html文件进行页面结构的布局并用css文件来设计页面的样式，以及使用js文件在页面中添加逻辑行为。而在微信小程序中也采用了相似的结构体系，即微信小程序每个页面的描述都是通过页面描述文件（以.wxml为后缀）、页面逻辑文件（以.js为后缀）、页面配置文件（以.json为后缀）以及样式表文件（以.wxss为后缀）这四个文件来进行的。

有了这些概念，当我们打开微信小程序的开发工具时，查看初始的目录结构可以看到，在项目的根目录下有2个子目录和3个文件。在根目录中的3个文件分别是app.json 、app.js 和app.wxss，他们构成了微信小程序系统的主体文件。pages目录下有2个子目录，分别是index和logs，他们其中都包含着4个相同名称的文件，分别以js、wxml、wxss、json为后缀名。每个子目录代表着小程序系统的一个页面。

(2)主体文件

每个小程序都是由根目录下的三个文件构成主体部分，他们是微信小程序的主体框架，用来对pages中各个页面提供支持，以及对逻辑层的调用，还有解析数据，wxss文件和wxml文件的功能。分别是app.js，app.json以及app.wxss。微信小程序在初始化运行时将会读取这些文件，并由此来生成小程序实例。

app.js，这是微信小程序的主逻辑文件，包含一些全局或公共的逻辑，主要是用来注册微信小程序。

app.json，这是微信小程序的全局配置文件，包含配置组成小程序的页面以及窗口的默认标题和背景颜色等，主要用来对微信小程序进行整体的配置。

app.wxss，功能类似于html中的css文件，是微信小程序的公共样式表文件，用来对页面的样式进行设计，需要注意的是在此文件中定义的样式，在别的任何子页面里也能够调用。

(3)页面文件

一个微信小程序中往往包含多个页面，需要注意的是在小程序中，所有的页面甚至首页都必须在pages文件夹下创建其子文件夹。项目中一共有多少个页面，在pages文件夹下就需要有对应的子文件夹个数。而在每个页面的子文件夹之中包含的四个文件的文件名必须相同，并经由四种不同的后缀名来进行区别，后缀名的含义如下：

.js文件是页面的逻辑文件，在其中进行控制页面的逻辑的JavaScript代码编写，用来实现此页面中需要与后端服务器接口进行的所有交互以及数据的请求。这个文件作为核心文件，是每个页面中都不可或缺的。

.wxml文件是页面的描述文件（类似网页中的html文件），负责设计与搭建当前页面的结构，以及数据的绑定，这也是每个页面必不可少的文件

.wxss 文件是页面的样式文件（近似于网页中的css文件），用来设计当前页面结构中需要的各种样式，例如字体的大小颜色等，，此外，在当前页面结构中还能够调用app.wxss中已经配置好的样式。同样也能够使用内联样式，如果当前页面也有相同的样式描述，由使用层叠样式的规则来决定最终的样式。当然如果此页面中需要使用到的样式都在app.wxss中定义好了，这个文件则可以省略配置

.json 文件是页面的配置文件，可以进行当前页面的窗口标题以及背景颜色等的配置，而且因为在app.json中已经有过全局的配置，如果在本页面中不需要进行改动，则可以省略此文件的配置。

(4)其他文件

除了上述的主体文件和页面文件，小程序中通常还会用到一些图片、视频或者音乐等的文件，这些资源类的文件也可以单独创建子目录来存放。当然，一般在微信小程序中也只需要保存界面所使用的一些静态图片就可以，而音频、视频这些大文件通常是从后端网站调用。除此之外，我们在小程序的根目录下还能看到一个名为utils的文件夹，里面存放着可以被调用的逻辑模板文件，相当于一个工具箱，在任意一个页面的js文件中都可以通过import该模板文件从而调用其中已经定义好的各种函数。

2.2逻辑层

虽然微信小程序在整个系统中属于表现层，但是在这个前段中通常也会对从后台收到的数据进行进一步的加工处理。而且，界面中的数据也可能会随着后台数据动态的更改而转变，因此必须在前端的页面中进行一些逻辑代码的编写。微信小程序这个前端系统也进行了层析的划分，分为逻辑层（App Service）和视图层（View）。

小程序开发框架的逻辑层包括根目录以及页面文件夹里的.js文件，文件格式遵循JavaScript语言规范，逻辑层负责小程序的数据处理，从视图层接收数据，处理完之后再把结果返回给视图层，同时接受来自视图层的反馈。

在小程序中要求使用app和pages方法来注册程序和页面，通过getApp方法来获取app方法的实例。在面向对象编程中，类的实例是经由对象来实现的，而JavaScript的对象是基于函数的，例如如小程序自身就是通过一个函数App（）实现的，小程序的页面则是通过Page（）函数实现的，因此我们可以认为，小程序的实例也就是构成小程序的各个函数。

用户在开发者工具中编写的全部逻辑层代码，在开发完毕后会被整合成一个JS文件，在小程序启动的时候，这个文件将被下载来运行，并且有离线执行的能力，直到小程序销毁。这种机制可以让代码做到离线使用，所以微信把逻辑层称之为AppService。

2.2.1小程序的初始化

官方对逻辑层使用的JavaScript进行了一些封装，在主逻辑文件app.js里定义了一个用来注册的App函数，需要注意的是，每个小程序在初始化过程中都必须在通过这个函数来实现注册。小程序的注册和初始化是通过App（）函数进行的，该函数的参数是一个JSON对象，在这个对象中可以初始化小程序的生命周期函数。一个对象的生命周期，便是从对象创建运行、提供服务、暂停终止服务、到最终被销毁的整个过程。

onLunch函数作用于小程序的初始化阶段，小程序的一个完整生命周期里onLunch函数只会触发一次。而onShow函数则是在小程序初始化或者从后台转入前台运行时触发，当小程序从后台切换到前台时会触发onHide函数。

对于小程序后台与前台运行的概念，当用户单击小程序的关闭按钮，或者按了手机的Home键离开微信，或者来电话离开了小程序页面时，小程序不是被直接销毁，而是转入后台运行状态，而在用户重新进入小程序页面时，小程序就会自动恢复至前台运行状态，而不需要重新启动加载。只有当进入后台一段时间或者系统资源占用过高时，才会被系统真正销毁，这个销毁时间是由手机操作系统（os）决定的。

在下面的App（）函数示例中，小程序初始化时指定了上述的三个生命周期函数和一个全局变量globalData：

```
App({
onLaunch: function () { //当小程序初始化完成的时候，触发这个函数},
onShow: function () { //当小程序启动，或者从后台切换到前台的时候，触发这个函数},
onHide: function () { //当小程序从前台进入后台的时候，触发这个函数},
user: { userInfo: 'Hello World'}
})
```

小程序框架中给出了一个在其他页面中也可以调用的函数，即getApp（），经由这个函数我们能够得到小程序的实例。以下所示代码就是在index.js中引用了app.js中的App（）实例：

```
//获取应用实例
const app = getApp()
console.log(app.user.userInfo)
```

需要注意的是，App（）必须且只能在根目录的app.js文件中注册并且只能注册一个。在App（）函数内需要拿到App实例时，只要使用this关键字就可以达到目的，不需要调用getApp（）。需要注意的是在经由getApp（）函数获得到实例以后

，不能擅自对生命周期函数进行调用，否则容易出现错误。

2.2.2页面的注册

在小程序里需要使用一个页面之前，需要先在Page（ ）函数中完成注册。这里的注册函数与之前的App（ ）函数类似，其中使用的参数同样是一个JSON对象，包含着页面中的初始化数据以及生命周期函数，例如data、onLoad、onReady、onShow等，还可以通过编写自定义的函数用来相应页面的事件。

下面给出了小程序中一个页面的index.js代码示例，其中包括了用Page（ ）函数完成了页面的初始化、生命周期的定义等：

```
Page({
data: { motto: 'Hello World',},
onLoad: function (e) { //页面初始化，e为页面跳转所带来的参数 },
onShow: function () { //页面显示时触发执行的操作 },
onReady: function () { //页面渲染完成时触发执行的操作 },
onHide: function () { //页面隐藏时触发执行的操作 },
onUnload: function () { //页面关闭时触发执行的操作 },
onPullDownRefresh: function () { //下拉刷新时触发执行的操作 },
onReachBottom: function () { //上划刷新时触发执行的操作 },
//事件处理
getUserInfo: function(e) {
this.setData({ motto: e.detail.userInfo })
} })
```

data中存放的是初始化数据，也正是通过data来渲染页面，渲染过程即是以JSON语法的形式将data数据从逻辑层送入视图层，因此data中数据的格式必须是JSON语法所支持的。

在Page（ ）函数中包含着五个生命周期函数。

onLoad函数只在页面加载的时候调用一次，可以接受到wx：navigateTo和wx：redirectTo事件以及 < navigator/ > 中的query事件中传递过来的参数；

onReady函数在页面第一次渲染完成时被调用，因此在每一个页面只有一次的调用机会，表示该页面现在已经准备就绪了，可以与视图层进行数据交互了。需要注意的是对页面的设置例如wx：setNavigationBarTitle等函数的使用都需要在onReady之后再进行。

onShow函数则是在页面显示时被调用，即在每次打开页面的时候都会进行调用；

onHide函数在页面隐藏的时候被触发，例如在用户通过点击小程序底部的导航按钮来进行页面的切换或者通过navigateTo实现页面导航时当前页面就会被隐藏。

onUnload函数在页面被卸载的时候被调用，例如在执行redirectTo或着navigateBack操作的过程中页面当前页面会被卸载而跳转至其他页面，此时就会调用onUnload函数。

在Page（ ）中不仅可以进行data数据的设置以及五大生命周期函数的定义，还能够声明其他的事件处理函数。在视图层中添加绑定事件，当摸个事件被激发时，Page（ ）函数中声明的事件处理函数就会被调用。如上面给出的getUserInfo函数。用来把设置好的数据从逻辑层传递至视图层的函数是setData，而当setData函数被调用时，data中的数据将被刷新。在setDate被调用时，会接收一个对象，通过key，value键值对的形式将data中key的对应值修改为value值。

2.2.3模块及调用

在JavaScript中声明的函数和变量只能在当前文件中使用，所以在不同的页面文件中声明相同的函数或变量名是可以的，并不会产生影响。为了提高代码的可读性以及有效利用率，我们可以对小程序的模块化特性加以利用，模块化指的是将重复的或者公用的代码从程序中提取出来，将其打包成一个单独的js文件，将其设定为提供特定的功能的工具包，一般都将这些js文件放置在utils文件夹下。目前，小程序的js代码模块能够利用module.exports方法将其接口对外开放。举例如下：

```
function speak(name){
console.log("hello"+name)
}
module.exports={
speak
}
```

在index.js中调用的时候，采用import方法即可将以上的js模板代码导入：

```
import { speak } from '../../utils/util';
```

2.2.4微信原生api

在微信小程序的开发框架之中，为广大开发者们提供了大量的微信原生API，让我们可以更加便捷地调用微信提供的能力，如获取登陆的用户信息、支持本地存储以及支付功能等。

目前小程序所出给的原生API包括网络、数据缓存、媒体、文件、位置、设备、界面、WXML节点信息、WXML节点布局相交状态以及开放接口这几大类，在微信小程序的官方文档中给出了每一类的详细介绍。

2.3视图层

小程序的视图层就是我们在手机上打开小程序看到的并且可以操作的视觉页面，视图层负责页面的视觉元素的组织和与用户的交互，视图层依靠数据通信和一系列的事件与逻辑层相连接。在微信小程序中视图层由wxml和wxss文件组合而成，wxml文件负责页面结构的布局，基于XML语言规范，也就是微信所说的WXML语言；wxss文件负责组件的视觉样式描述，基于CSS语言规范，也就是微信所说的WXSS语言。同时微信也提供了丰富的视觉组件来进行页面的内容组织和展示，视觉组件类似HTML语言的标签，是构成小程序页面的基本用户界面元素。

2.3.1 WXML语言

WXML是WeiXin Markup Language的简写，翻译过来就是“微信标记语言”，是微信定义的一款进行页面结构布局的语言，在搭建小程序页面的过程中还需要结合微信所供给的视图组件以及各种事件进行使用。

在小程序框架中，将视图层和逻辑层之间的交互方式称之为事件（event），通过事件，能够把用户在视图层的行为活动反映给逻辑层，在逻辑层接收到之后再将处理的结果返回给视图层。在wxml页面文件中，事件处理函数往往绑定在视图层的组件上，例如bindtap，当用户点击到该组件时，事件会被触发，从而调用逻辑层中对应的事件处理函数。

小程序里的视图组件也就相当于HTML文件中的“标签”，而组件也正是WXML语言最基础的视觉单位，在小程序框架中，供给了丰富的基础组件给开发者们使用，让开发者可以通过将这些基础组件组合起来，从而快速地实现想要的功能。

数据绑定是视图层数据与逻辑层数据通信的方法，也就是将一个用户的界面元素（组件）的属性绑定到一个逻辑文件的对象实例上的方法。从视图组件的角度来看，数据绑定这种机制其实就是把程序绑定到用户界面。需要注意的是，页面中的动态数据全都是来自对应地页面逻辑文件中的data。

渲染，也就是我们利用WXML代码在小窗口页面上生成可以与用户交互的图像的过程。在WXML中，我们用wx：if = “{|condition|}”控制属性来进行判断是否需要渲染一个代码块，当wx：if绑定的值为真（true）时开始渲染，这就是条件渲染。有时候我们需要对一个组件重复进行渲染，就可以使用wx：for这一组件可以帮我们效率更高的完成这项工作，wx：for将会绑定一个数组，然后重复使用数组中各项数据来渲染组件，即列表渲染。

在WXML可以通过使用模板（template）来定义一个代码片段，这样在其他的WXML文档中就可以使用该模块片段。如果某几个组件的组合要反复使用到，这时可以考虑将这些组件的组合定义为一个模板，然后就可以在wxml中直接使用这个模板了。定义模板时，需要使用 < template > 标签，模板的名字由name属性设置，然后在模板中使用小程序的组件进行定义，使用方法与直接使用组件类似。在需要使用该模块的时候用is属性来指定调用的模板名称，data属性将所需要的数据传入模板中，就相当于是模块的作用域。

当我们需要在一个WXML文件中使用另外一个WXML文件定义的模块时，就需要使用WXML语言的引用语句，经由import和include这两种方式就能够引用其他文件的内容。如果被引用的文件定义了模板代码，则需要使用import方式进行引用。而使用include可以将源文件中除了模板定义之外的其他代码全部引入，其引入方法想防御将源文件中的代码拷贝到include所在的位置。

2.3.2 WXSS语言

在视图层中不仅有页面结构文件WXML，通常还要配合使用页面样式文件WXSS。WXSS是WeiXin Style Sheets的简写，翻译过来就是微信样式表，是微信基于CSS语法规范而定义出的一种新的样式语言，可以对WXML文件中组件进行样式的设计与修饰，包括背景，文本，字体，链接等方面的样式。虽然在WXML中也能够为各种组件添加样式，但文件结构会显得过于臃肿，不符合代码的书写规范。而在WXSS中则可以更加规范，高效地进行样式修饰。所以通常WXML文件只用来进行页面结构的布局，而WXSS则用来修饰页面结构的样式。在小程序中，WXSS拥有CSS的大部分特性，包括选择器的支持以及使用class和style方式设置组件的样式。

在微信小程序中可以通过内联样式以及外部样式表的方法来使用样式表文件。内联样式只对单个组件的渲染有效，常用于精确的控制某个WXML视图组件的样式定义，以及从逻辑层接收动态的样式改变，小程序的组件支持使用style、calss属性使用内联样式。小程序还支持用@import语句来进行外联样式表文件的调用，使用时将其相对路径写在@import之后。小程序提供了样式表独立于WXML内容文件的框架，以.wxss文件名的形式存在于小程序根目录和页面目录下。定义在根目录app.wxss文档中的样式可以适用于小程序的全部页面，称之为全局样式。定义在page页面目录下的.wxss文件只适用于于该页面本身，称之为局部样式。局部样式被页面使用时会覆盖掉全局样式中相同选择则其的内容，也就是说在页面显示时，局部样式的优先级更高。

微信在定义CSS语言规范时，继承了CSS语言几乎所有的特性，但由于微信小程序页面主要作用于在手机及其他移动设备的屏幕上显示，WXSS拓展了CSS在显示尺寸单位上的定义。微信在CSS像素（px）的基础上，又进行了概念上的拓展，为了让小程序更加适应屏幕尺寸和分辨率的碎片化，增加了rpx（响应式像素：能够根据屏幕宽度的不同而进行自适应调整的像素单位）作为屏幕尺寸的一个定义标准。

第3章音乐播放器功能设计与开发

3.1界面设计

本次项目所设计的播放器包含以下界面，分别是

1.首页，由四大模块组成，分别是顶部的搜索框模块，中部并列的新歌推荐和热歌排行榜模块，以及底部的音乐播放控制器。

（1）搜索框，如图4-1所示，位于首页的顶部，用户可以在此输入要查询的歌曲名或歌手名等关键字后，会自动查询出相关结果显示在搜索展示界面，点击取消按钮后会退出搜索功能重新回到原首页。

（2）新歌推荐，如图4-1所示，位于首页的中部，一共包括10首推荐的新歌，每一首歌都显示着音乐封面以及音乐名信息，点击任意一首歌曲将会开始播放同时跳转至音乐播放界面，在底部的音乐播放控制器和音乐播放界面都可以查看到当前的歌曲信息。

（3）首页热歌排行榜，如图4-2所示，位于首页的中部，排列着200首当前最流行的音乐，每一首都显示着歌曲序号，歌曲封面，歌名及歌手名等信息，排行前三首歌的歌名设定为红色字体，看起来比较醒目，点击歌曲将进行播放并跳转至音乐播放界面。

（4）音乐播放控制器，如图4-2所示，位于首页的底部，在左侧部分用户可以在此处看到正在播放的歌曲的封面，歌曲名及歌手名等信息，通过右侧的播放暂停和下一首按钮来控制音乐的播放状态及跳转。当点击左侧的音乐信息区域时将会跳转至音乐播放界面。

2.搜索展示界面，在首页进行歌曲搜索之后此页面将会出现，其中列表排列着搜索出来的相关结果，如图4-3所示，点击搜索结果其中的任一歌曲将会进行播放并跳转至音乐播放界面，此时点击返回按钮将回到之前的列表展示界面，可以继续查看其余的搜索结果。

3.音乐播放界面，在首页或者搜索结果页面中点击想要播放的音乐，以及点击首页底部的音乐播放控制器之后，就会跳转至此页面，从上至下的布局分别是歌曲封面的展示，歌曲名，歌手，相应歌词的显示以及底部的音乐播放进度条，如图4-4所示。

图4-1 首页-新歌推荐图4-2 首页-排行榜

图4-3 搜索结果图4-4 播放音乐

3.2网易云音乐API

要开发一款音乐播放器小程序，首先要考虑的就是音乐的数据来源。有了音乐的来源，才可以根据我们的需要在微信小程序中进行掉用和展示。所以说，微信小程序其实就相当于用来显示歌曲信息的前端平台，而后端系统就是音乐API。在网上我们可以找到许多免费的音乐接口，这样就不需要我们再次进行后端系统开发了，在这里我们选取了网易云音乐NodeJS版API进行使用。

先进行Node.js的环境安装与配置，选择相应电脑系统型号版本的安装包进行下载安装，完成后在我们选择下载的NeteaseCloudMusicApi中进行npm的安装，最后运行Node指令$ node app.js，即可启动服务器，注意服务器的启动默认端口为 3000，也可以通过指令进行修改。现在我们就可以使用网易云音乐所提供的各种接口了。

（1）网易云热歌排行榜

网易云热歌排行榜的接口地址是https://neteasemusic.leanapp.cn/top/list?idx=?

调用此接口时，需要传入参数 idx, 可以得到不同的排行榜，如"0": 云音乐新歌榜,"1": 云音乐热歌榜,"2": 网易原创歌曲榜等

调用该接口之后将返回一个JSON格式的数据，在result对象中包含着关于网易云音乐排行榜的详细信息，包括updateTime：排行榜的更新时间，coverImgUrl：当前排行榜的封面地址，description：当前排行榜的内容介绍等属性，其中tracks数组里的存放着当前排行榜的音乐列表，一共有100个元素，数组中每个元素里记录着每一首歌曲的详细信息，包括name：歌曲名 ，id：歌曲id ，artists数组：歌手信息 ，album对象: 专辑信息，在本次项目中我们需要使用到的数据主要就是以上这些，要使用这些数据时需要先定义一个数组例如temp[]存放result.tracks的值,然后遍历temp数组，分别通过数组中每一项item取出所要的数据信息，其中歌曲名：item.name，歌曲id：item.id，歌手名：item.artists[0].name，歌曲封面：item.album.picUrl.

（2）新歌推荐

新歌推荐的接口地址是：https://neteasemusic.leanapp.cn/personalized/newsong

调用此接口时无需传入参数将返回一个JSON格式的数据，其中result数组里存放着10组推荐的歌曲信息，包括name：歌曲名 ，id：歌曲id ，song对象：歌曲详细信息等，其中song对象里又包含着artist数组，存放着歌手信息，如歌手名和歌手id等，以及album数组，存放着音乐的专辑信息，如歌曲封面和发行公司信息等。在本次项目中我们需要使用到的数据主要就是以上这些，要使用这些数据时需要先定义一个数组例如temp[]存放result的值,然后遍历temp数组，分别通过数组中每一项

item取出所要的数据信息，其中歌曲名：item.name，歌曲id：item.id，歌手名：item.song.artists[0].name，歌曲封面：item.song.album.picUrl.

通过以上两个接口我们就可以把音乐播放器的首页展示所需要的数据获取到了。

（3）根据歌名或者人名查询歌曲

歌曲查询的接口地址：https://neteasemusic.leanapp.cn/search?keywords=?

调用此接口时需要传入参数keywords：关键词，可以搜索音乐和歌手，关键词也可以有多个，之间用空格分开。除了必选参数keywords之外，还有可选参数limit：返回数量（默认为30个），offset：偏移数量（默认为0），用于分页，使用公式为：（页数-1）*limit，，type：搜索种类（默认为1，即单曲）

调用该接口之后将返回一个JSON格式的数据，在result对象中包含着当前搜索结果的信息，包括一个songs数组：搜索出来的相关歌曲的集合以及songCount属性：搜索结果的总条数。其中songs数组里的存放歌曲数目由limit参数控制，数组中每个元素里记录着这条搜索结果的详细信息，包括name：歌曲名 ，id：歌曲id ，artists数组：歌手信息 ，album对象：专辑信息，在本次项目中我们需要使用到的数据主要就是以上这些，要使用这些数据时需要先定义一个数组例如temp[]存放result.song的值,然后遍历temp数组，分别通过数组中每一项item取出所要的数据信息，其中歌曲名：item.name，歌曲id：item.id，歌手名：item.artists[0].name，歌曲封面：item.album.picUrl.

（4）获取歌词

搜索歌词的接口地址：https://neteasemusic.leanapp.cn/lyric?id=?

使用此接口时需传入参数id即歌曲id，就能够获得对应音乐的歌词。

调用该接口之后将返回一个JSON格式的数据，在lrc对象中包含着当前搜索的歌曲的歌词信息，我们主要使用到的就是其中lrc对象中的lyric属性，lyric是一个字符串类型的数据，其中包含了每一句歌词及其出现的时间节点，它的格式举例如下："[01:00.880]我可以跟在你身后\n[01:04.130]像影子追着光梦游\n"。要取出这些数据并不困难，只需要通过lrc.lyric即可获得该字符串。但是要将其中每一句歌词取出并让其在相应的时刻出现在音乐播放界面的歌词展示部分，则需要对此字符串进行加工，这一部分的内容放在配置文件中进行详细讲解。

（5）获取音乐

获得音乐播放地址的接口地址：https://neteasemusic.leanapp.cn/music/url?id=?

调用此接口时需要传入参数id：歌曲id，此外还有可选参数br：码率（默认设置为最大码率即999000，如果想要320k码率，则可以设置为320000）

调用该接口之后将返回一个JSON格式的数据，在data数组中包含着当前获取到的歌曲的信息，如size：歌曲内存大小，type：歌曲的格式，url：当前歌曲的在线播放地址，我们主要使用到的就是其中的url属性，通过data[0].url即可获得。

3.3创建项目

在经过上述界面的设计构想以及对将要使用的网易云api接口的了解之后就要开始进入实际项目的开发过程了。

3.3.1创建新项目

首先打开app.json文件，对本小程序中的页面进行设置，在pages数组中添加3个页面文件夹的地址，分别是pages/index/index（首页），pages/now/now（音乐播放页面），pages/logs/logs（日志文件页面）。并对小程序的导航栏进行设置，在window数组中将导航栏标题中的文字内容修改为music："navigationBarTitleText": "music"；将导航栏背景颜色设置为绿色："navigationBarBackgroundColor": "#35b558"；将导航栏标题颜色设置为白色："navigationBarTextStyle":"white"；将下拉loading的样式设置为暗色。

在pages中添加了三个页面，其中的第一个默认设置为首页。在window属性中设置的音乐播放器的导航栏中标题的文字以及背景颜色等属性，是整个音乐播放器小程序的全局配置，会显示在播放器的每个页面上端。

3.3.2创建配置文件

因为在项目中要使用到网易云音乐API，且这个API的接口地址在多处要用到，所以最好是将这些内容封装在外部文件中，需要使用时直接引入该模块即可。

在项目的utils文件夹中创建文件net.js，首先定义一个常量preUrl将网易云API的地址https://neteasemusic.leanapp.cn保存其中。在这个文件中我们将声明一个方法apiRequest，将访问各大接口需要的参数如method，data以及各大接口的补充地址和访问成功或者失败时的success，fail方法都封装进了apiRequest方法中，然后通过微信小程序提供的网络api接口wx.request，对组合拼接起来的服务器最终接口地址发起请求，并将结果进行返回。最后通过module.exports声明此模块对外暴露apiRequest方法。在其他页面导入net.js文件之后，不管是获取新歌推荐还是排行榜，在调用apiRequest方法只需要传入该接口需要补充的地址及各种数据参数和success，fail方法即可。核心代码如下：

```
function apiRequest({url，method，data,，success, fail}){
return wx.request({url: `${preUrl+url}`,method，data，success，fail,})
}
module.exports= {apiRequest}
```

其次因为在处理获取到的歌词的时候需要进行String字符串的裁剪拼接，所以也可以单独将歌词的处理方法进行模块的封装。

在项目的utils文件夹中创建名为lyric.js的文件，首先通过import将上述的apiRequest方法导入，声明一个方法getlyric，需要传入参数歌曲id。通过调用apiRequest方法传入歌词获取的接口地址url: '/lyric'并设置method方法为GET，访问歌词获取接口，返回成功的话执行success方法，否则执行fail方法。在success方法中，先要对返回的数据进行判断，因为有可能出现网页繁忙提示503错误等，虽然还是有数据返回但已经不是我们需要的json数据了，此时仍然按照正常的json数据进行处理则会报错。还有一种情况就是这首歌曲并没有歌词，如果不经过判断就直接调用数据也会产生错误。针对上述两种情况，我们分别对suc.data.lrc以及suc.data.lrc.lyric进行验证，判断他们是否存在，如果不存在则直接返回false，不继续往下执行。判断成功之后将获取到的歌词字符串通过split('[')方法进行拆分，定义一个数组timeArr用来存放拆分之后的数据。通过遍历timeArr数组，再次对每个元素进行拆分，定义一个新数组res，将时间与歌词拆分开来并将标准的分秒时间格式转换为秒数，以此作为res数组的下标，将同时拆分出来的歌词存入其中，就实现了时间与歌词的同步对应。这样只要在播放音乐的时候不断调用res数组，看当前时刻是否是res数组中的下标，如果是则将对应的歌词取出展示在音乐播放界面。其核心代码如下：

```
let time = suc.data.lri.lyric.split('[');
let res = {};
time.forEach((item) = > {
let index = parseInt(item.split(']')[0].split(':')[0]) * 60 + parseInt(item.split(']')[0].split(':')[1]);
let value = item.split(']')[1];
res[index] = value;})
```

同时还需要一个将秒数换算成标准分秒显示格式的formatSeconds方法，首先将秒数对60取余，得到的值就是秒位置的数值，并进行判断其是否小于10，如果是为了规范化则在显示时要在前面补一个0，至于分钟数则采用Math.floor对60做除并向下取整，同样为了规范化需要与10作比较，最后将分钟数与秒数通过“：”进行组合形成一个新的字符串。以下是核心代码：

```
sec = s % 60;
min = Math.floor(s / 60) % 60;
t = min + ":" + sec;
```

最后同样需要将上述两个方法模块化，module.exports = {getlyric，formatSeconds,}，这样两个工具文件就准备好了。

3.2.3导入图标

在之前的音乐播放器界面展示过程中可以看到在本次项目中会出现一些图片和小图标，其中的歌曲封面是通过音乐接口api动态获取的，而在初次进入音乐播放器首页时，底部音乐播放控制器的的歌曲封面位置应当采用音乐cd图标来进行填补，其他使用到的一些音乐播放器相关的小图标，包括音乐播放控制器的播放，暂停，下一首按钮等，应当提前准备好，本项目中使用到的小图标如图4-5所示。

图4-5 音乐播放器小图标

到此为止我们开发前的所有准备工作就已经完成了，接下来就要开始着手于各页面的功能开发。

3.4首页界面搭建及功能开发

3.4.1开发页面文件

对首页界面从上往下进行线性布局 ，首先对搜索框模块进行开发。在进入首页之后我们可以看到搜索框的样式如图3-1所示，在点击搜素框之后样式则变为图3-2。

图3-1

图3-2

打开index文件夹里的index.wxml，对搜索框模块进行编写，为了页面的美观性，在搜索框这里的布局样式直接调用已有的weui.wxss样式文件。首先是图3-1中展示的搜索框，其中只包含一个icon组件和文字部分，通过label组件将他们组合起来，为了进行上述两个视图之间的切换，需要在label中设置hidden属性并添加一个绑定事件showInput，通过对inputShowed这个属性值的判断决定页面上显示哪一个视图，点击搜索框将触发事件，在逻辑层中对inputShowed的值进行更改从而使图3-1隐藏并将图3-2显示出来。接下来是图3-2中的搜索框，包含左端的icon组件，中部的input组件以及最右端的一个取消按钮，在取消按钮中同样包含着hidden属性并绑定了事件，可以通过点击取消按钮在逻辑层中再次对inputShowed的值进行更改，从而使当前视图隐藏切换至图3-1，也就是退出了歌曲搜索。在input组件中通过绑定事件inputTyping将输入的关键词实时传送至逻辑层并进行歌曲的搜索。需要注意的是在输入关键词后在input组件的右部，取消按钮的左部之间会出现一个叉叉icon图标，如图3-3所示，在这个组件中添加了wx:if属性来对当前输入框内容的长度进行判断并绑定了clear事件，点击图标将清空输入框。

图3-3

在进行关键词的输入之后，会触发input组件中的inputTyping事件，自动进行歌曲的搜索功能，此时下方页面将会对搜索结果进行列表展示。因为搜索的结果一个页面肯定显示不完，因此我们使用滚动视图scroll-view组件将搜索结果进行展示，设定lower-threshold的值为20，即上拉至距离底部20px时触发scrolltolower事件，重新调用搜索方法，增加查询页数。首先还是要通过wx:if属性对输入框是否有内容进行判断: wx:if="{{inputVal.length > 0}}"，如果判断成功则将从逻辑层反馈回来的歌曲搜索结果searchResult数组进行列表渲染，通过wx:for将每一条搜素结果都置于navigator组件中，并把其中的歌曲名和歌手名信息取出排列在页面中，如图3-4所示。通过navigator组件可以在点击搜索结果列表中的任意一条时跳转至音乐播放界面，还要

同时进行歌曲的播放，则需要在navigator组件中添加绑定事件toPlay，并将当前点击歌曲的信息存放在navigator组件中的属性中，如data-author:歌手名，data-id:歌曲id，data-picUrl:歌曲封面，data-name:歌曲名。在触发事件之后在逻辑层会把这些歌曲信息保存至本地缓存，到歌曲播放界面再取出来，同时通过歌曲id查询到当前音乐的播放地址，通过wx:playBackgroundAudio接口进行歌曲的播放。还需要设置open-type跳转方式为navigate，这样在进入歌曲播放页面之后仍然可以返回至搜索结果展示页面。

图3-4

至此已经处理完成了搜索页面的开发，接下来对新歌推荐和排行榜模块进行布局。

首先是标题的设计，通过currentTab的值来表示当前选中的模块是推荐音乐还是排行榜，在index.js中设定默认值为0，即我们打开音乐播放器页面中显示的是新歌推荐内容，设定样式使选中的模块标题颜色变为绿色，如图3-5所示。并在组件中添加绑定事件swichNav，当点击标题时将会在逻辑层中对currentTab的值进行修改，从而实现对新歌推荐与排行榜页面的切换。

图3-5

接下来进行内容的布局，因为这两个模块是并列的所以在这里采用swiper组件来进行他们之间的滑动切换，并添加current属性对上述的currentTab属性值进行确定页面中展示的是推荐音乐模块还是排行榜模块，并在组件中添加绑定事件bindChange，在进行页面滑动时将会在逻辑层中对currentTab的值进行修改，从而实现对新歌推荐与排行榜页面的切换。

继续编写新歌推荐和排行榜的内容展示页面，首先进行新歌推荐的页面代码编写，将从逻辑层获取到的推荐新歌song数组进行列表渲染，通过wx:for将其中的每一首歌曲信息都置于navigator组件中，并把其中的歌曲封面和歌曲名信息取出排列在页面中，如图3-6所示。通过navigator组件可以在点击想要播放的歌曲时跳转至音乐播放界面，还要同时进行歌曲的播放，则需要在navigator组件中添加绑定事件toPlay，并将当前点击歌曲的信息存放在navigator组件中的属性中，如data-author:歌手名，data-id:歌曲id，data-picUrl:歌曲封面，data-name:歌曲名。在触发事件之后在逻辑层会把这些歌曲信息保存至本地缓存，到歌曲播放界面再取出来，同时通过歌曲id查询到当前音乐的播放地址，通过wx:playBackgroundAudio接口进行歌曲的播放。还需要设置open-type跳转方式为navigate，这样在进入歌曲播放页面之后点击返回依然是当前浏览的新歌推荐的位置。

接下来进行排行榜页面代码编写，和新歌推荐的页面结构大致相同，只是将从逻辑层获取到的数据替换为排行榜tracks数组，依然是进行navigator组件的列表循环，以及绑定同样的事件toPlay，在点击歌曲时将进行播放并跳转至音乐播放界面。有所不同的是在排行榜的展示过程中不仅有每首歌的封面歌曲名及歌手信息，还添加了每首歌曲的序号，并将前三首歌曲名设定为醒目的红色，如图3-7所示。这就需要在遍历数组的时候进行条件渲染，通过wx:if将前三首歌曲判断出来:wx:if="{{index < 3}}"并通过wx:else对其他歌曲正常处理。

图3-6 图3-7

完成上述界面的开发之后，现在进行首页的最后一个模块，即底部的音乐播放控制模块的布局，包括左部的歌曲信息模块以及右部的播放控制按钮，如图3-8所示。在左部的歌曲信息模块里包含着当前播放音乐的封面，歌曲以及歌手名，分别通过audio的三个属性值audio.picUrl，audio.name，audio.author即可获取到对应的数据信息。在初次进入首页时由于没有播放的歌曲，歌曲封面组件里应当传入之前准备好的图片地址，如图3-7所示。并绑定skip事件进行点击跳转音乐播放界面的操作。在右部的歌曲播放控制按钮组件中，分别引入之前准备好的音乐播放，暂停，下一首的图标的地址，并添加绑定事件audioPlay，audioPause以及audioNext，在点击相应的图标时分别完成播放播放，暂停以及下一首的功能。需要注意的是，播放和暂停的图标组件中需要设置hidden属性，通过对isplaying属性值的判断确定当前的播放状态，并进行图标的切换，所以在audioPlay，audioPause事件中还需要对isplaying的值进行修改。

图3-7 图3-8

至此首页的页面开发就完成了，接下来进行逻辑代码的编写。

3.4.2开发页面逻辑代码

首先通过import将之前net.js中准备好的的apiRequest方法导入，import { apiRequest } from '../../utils/net'，并定义一个全局变量page=1，在歌曲搜索功能里会使用到。然后在Page中对首页的所有需要用到的数据进行进行初始化配置，data中的数据包括winHeight: 当前手机的屏幕高度，currentTab: tab切换，inputShowed: 输入框显示状态，inputValue:输入框内容，searchResult: 搜索结果，newsong: 推荐新歌，tracks: 排行榜，isplaying: 当前音乐播放状态，以及一个audio对象通过id，name，src，picUrl，author这几个属性包含了歌曲的所有信息。

然后在监听页面加载onLoad生命周期方法中调用apiRequest方法分别进行推荐音乐及排行榜的获取，并将所得数据传入data中。

在获取推荐新音乐过程中，通过调用apiRequest方法传入新歌获取的接口地址url: '/personalized/newsong '并设置method方法为GET，访问新歌获取接口，返回成功的话执行success方法，否则执行fail方法。在success方法中，定义一个temp数组用来存放获取到的推荐新歌，在前面的网易云API介绍中我们已经知道返回的歌曲数据存放在suc.data.result数组中，通过遍历suc.data.result数组即可取出我们需要的数据信息，对于数组中每一项元素item，歌曲id是item.id，歌曲名是item.name，歌手是item.song.artists[0].name，音乐封面是item.song.album.picUrl，每首歌的信息组合成一个对象存放在temp数组中，最后把temp数组赋值给data中定义的newsong。

在fail方法中，将返回的错误信息打印到控制台，以便我们对错误进行排查。核心代码如下：

```
suc.data.result.forEach((item, index) = > {
temp.push({ id: item.id， name: item.name,
author:item.song.artists[0].name, ptr: item.song.album.picUrl});});
}); that.setData({ newsong: temp });
```

在获取排行榜过程中，通过调用apiRequest方法传入排行榜的接口地址url: '/top/list '并设置method方法为GET，同时还需要传入数据参数idx:1，表示我们获取到的是云音乐热歌榜，访问音乐排行榜接口，返回成功的话执行success方法，否则执行fail方法。在前面的网易云API介绍中我们已经知道返回的歌曲数据存放在suc.data.result.tracks数组中，定义一个dataTracks数组存放歌曲的信息。通过遍历suc.data.result.tracks数组即可取出我们需要的数据信息，对于数组中每一项元素item，歌曲id是item.id，歌曲名是item.name，歌手是item.artists[0].name，音乐封面是item.album.picUrl，将每首歌的信息组合成一个对象存放在temp数组中，最后把dataTracks数组赋值给data中定义的tracks。同样的在fail方法中打印错误信息。以下是核心代码:

```
suc.data.result.tracks.forEach((item, index) = > {
dataTracks.push({ id: item.id, name: item.name,
author: item.artists[0].name, picUrl: item.album.picUrl }) });
});that.setData({ tracks: dataTracks });
```

排行榜的数据获取与推荐音乐大致相同，不过还需要通过wx.setStorage将获取到的排行榜数据存储在本地缓存中，因为在播放下一首歌的绑定事件中会再次使用到排行榜信息，只需要使用wx.getStorage即可取出数据。wx.setStorage通过键值对的方式进行存储，示例:wx.setStorage({ key: 'tracks', data: dataTracks,})

除了上述的推荐音乐和排行榜数据的获取，还有的关键性代码就是搜索功能的实现，在Page中添加事件处理函数，也就是搜索框中绑定的inputTyping事件。当搜索框中有输入时，触发此事件，通过e.detail.value将输入内容获取到并

赋值给data中定义的inputValue。通过调用apiRequest方法传入歌曲搜索的接口地址url: '/search'并设置method方法为GET，同时还需要传入数据参数，包括keywords:关键字，offset:偏移数量以及limit:加载数目，设置为在第1页查询出15条搜索结果进行展示。返回成功的话执行success方法，否则执行fail方法。在前面的网易云API介绍中我们已经知道返回的歌曲数据存放在suc.data.result.songs数组中，定义一个result数组存放搜索结果信息。通过遍历suc.data.result.songs数组即可取出我们需要的数据信息，对于数组中每一项元素song，歌曲id是song.id，歌曲名是song.name，歌手是song.artists[0].name，音乐封面是song.album.picUrl，将每首歌的信息组合成一个对象存放在result数组中，最后把result数组赋值给data中定义好的searchResult。同样的在fail方法中打印错误信息。最后同样也要通过wx.setStorage将获取到的搜素结果数据存储在本地缓存中。核心代码如下:

```
suc.data.result.songs.forEach((song, index) = > {
result.push({ id: song.id, name: song.name,
picUrl: song.album.picUrl, author: song.artists[0].name });
});that.setData({ searchResult: result});
```

而在搜索结果展示页面中当我们往上拉到底部时会触发searchResultLower事件，此事件是用来更新之前的搜索结果，该事件处理函数与上述函数基本类似，不过page会以自增的方式赋值给offset,这样就可以不断获得新的搜索结果数据。同时在获取到新的搜索结果数组之后还需要对data中定义好的searchResult进行更新。并再重新使用wx.setStorage将本地缓存中的数据更新。核心代码如下:

```
result = Object.assign(that.data.searchResult, result)
that.setData({ searchResult: result});
wx.setStorage({ key: "searchResult", data: result})
```

接下来处理点击歌曲时触发的toPlay事件，定义一个songData对象来存放当前点击的歌曲信息，因为之前在navigator组件中添加了歌曲信息的各个属性，所以分别通过e.currentTaget.dataset中的id属性获取到歌曲id， name属性获取到歌曲名，picurl属性获取到歌曲封面，author属性获取到歌手，并存入songData中。然后通过使用wx.setStorageSync将songData存入本地缓存，等到音乐播放界面再将其取出。通过调用apiRequest方法传入音乐获取的接口地址url: '/music/url'，并设置method方法为GET，同时还需要传入数据参数：歌曲id:songData.id。返回成功时执行success方法，将歌曲的信息存入在data中定义的audio对象，其中歌曲的播放地址由之前的网易云API介绍可知是res.data.data[0].url，将其赋值给audio中的src属性。此外还需要将data中的isplaying的值设置为true，表示歌曲正在播放。紧接着通过微信小程序官方提供的wx.playBackgroundAudio进行音乐的后台播放，需要传入的参数分别是dataUrl:歌曲的播放地址，title:歌曲名，coverImgUrl:歌曲封面，这样歌曲就顺利播放出来了。同样在返回失败时执行fail方法，将错误信息打印到控制台进行查看。

接下来实现音乐的播放与暂停功能，在点击图标时将分别触发audioPlay和audioPause这两个事件。在audioPlay方法中，通过将audio中的src:音乐播放地址，name:歌曲名以及picUrl:歌曲封面这些属性值传入微信小程序提供的媒体API接口wx.playBackgroundAudio从而进行音乐的后台播放，同时将isplaying的值置为true。在audioPause方法中，通过微信小程序官方提供的wx.pauseBackgroundAudio()方法暂停后台音乐的播放，并将isplaying的值置为false。

接下来实现播放下一首音乐的功能，在点击下一首图标时将触发audioNext事件。之后将播放的是当前歌曲在排行榜中下一首，所以在audioNext方法中要先通过wx.getStorage将当前播放的歌曲信息clickdata以及之前存入本地缓存的排行榜信息tracks取出，并通过findIndex方法在其中进行歌曲id的对比，在排行榜数组中查找出当前歌曲的下标，即在排行榜中的名次，将获取到的名次加1就是下一首歌曲了。而如果当前歌曲不在排行榜中，则将播放排行榜中的第一首歌曲。接下来通过获取到的歌曲id进行音乐播放地址的查询并通过微信小程序提供的wx.setStorageSync接口更新clickdata当前歌曲信息的本地缓存，最后再次使用wx.playBackgroundAudio进行当前歌曲的后台播放。核心代码如下:

```
wx.getStorage({ key: 'clickdata',
success: function (clickdata) {
wx.getStorage({ key: 'tracks',
success: function (track) {
let currentSongIndex = track.data.findIndex((item) = > {
return item.id == clickdata.data.id;
}) currentSongIndex++;
```

到此为止首页的开发就全部进行完毕了，接下来进行音乐播放界面的开发。

3.5音乐播放界面搭建及功能开发

3.5.1开发页面文件

在推荐音乐，排行榜以及搜索结果列表里，点击任一首歌曲，都将导航至音乐播放界面，接下来就来进行音乐播放界面的开发。在上图中已经展示过音乐播放界面，整体结构比较简约，在上方显示歌曲的图片，图片下方是音乐名和歌手，以及按时出现的对应歌词，最下方是音乐播放器的进度条。

打开now文件夹下的index.wxml文件，首先是对整个音乐播放界面的背景处理，将通过audio.picUrl获取到的歌曲封面添加滤镜，并设置高斯模糊，以此作为整个界面的背景。然后对封面圆圈模块进行设置，在组件样式中对rotate的值进行判断，如果为true则添加rotate样式让圆圈里的封面进行旋转，否则停止旋转，rotate的值由歌曲的播放和暂停状态确定。接下来是歌曲的三大信息展示，分别通过audio.name，audio.author以及lrc获取到歌曲的名称，歌手以及歌词信息。接下来进行音乐进度条的代码编写，使用slider组件，配置其中的value以及max属性，分别获取到当前音乐的播放进度和总时长，这样就能够动态的展示进度条的变化，在进度条下方还需要将已经播放的时长和总时长的标准分秒格式展现出来。在组件中还需要绑定事件sliderChange，将拖动进度条后的位置传入逻辑层，通过wx:seekBackgroundAudio对value重新赋值，从而改变当前音乐播放的进度。

3.5.2开发页面逻辑代码

首先通过import导入之前net.js中准备好的的apiRequest方法，以及lyric.js中的getlyric和formatSeconds方法。然后在Page中对音乐播放界面的所有需要用到的数据进行进行初始化配置，data中的数据包括value:当前音乐所播放时长秒数，max:音乐的总秒数，time: 当前音乐播放时长的标准格式，alltime: 音乐总时长的标准格式，rotate: 是否旋转，lrc: 歌词，以及一个audio对象通过id，name，picUrl，author这几个属性包含了歌曲的所有信息。需要注意的是对于音乐播放器的进度条显示，使用time和alltime两个字符型变量进行展示，而value和max其中存放的是实际的秒数。

接下来需要在监听界面加载函数onLoad中，通过小程序官方提供的原生api接口wx:getStorage获取到本地缓存中的当前播放音乐信息clickdata，并将其中的歌曲id，歌曲名，歌曲封面以及歌手分别赋值给audio对象中的id，name，picUrl以及author属性。

因为要动态的加载所获取到的歌词以及音乐播放的进度，需要在监听页面显示函数onShow中设计一个播放器方法player并每隔一段时间来调用该方法，在player方法中先通过wx.getBackgroundAudioPlayerState方法获取到后台音乐的播放状态，其中的属性包括currentPositions: 选定音频的播放位置，duration: 选定音频的长度（单位：s），status: 播放状态（2：没有音乐在播放，1：播放中，0：暂停中）,将currentPosition和duration的属性值分为赋给data数据中的value和max。然后通过之前引入的formatSeconds方法将上述两个值转换为标准的分秒时间格式，并分别存入time和alltime字符串中。还需要对status的值进行判断，若等于1即处于播放状态时，将rotate的值设置为1，也就是让封面圆圈进行旋转。接下来在player方法中将对歌词进行处理，通过之前引入的getlyric方法，将当前歌曲的id传入，在返回的数组中以当前的播放进度位置为下标判断是否存在对应的歌词，如果存在就将歌词传入data中的lrc。最后通过setInterval设置此播放器的定时刷新时间，以获得当前进度条的实时变化及歌词的动态显示。核心代码如下：

```
this.player = function () {
getlyric(that.data.audio.id, (obj, lyricArr) = > {
if (obj[currentPosition])
that.setData({ lrc: obj[currentPosition]})
```

}} this.Interval = setInterval(this.player, 1000);

然后来进行音乐播放器进度条的滑动功能开发，在进度条组件中绑定事件sliderChange，通过触发事件将拖动的进度条当前位置送入sliderChange函数中，然后使用wx.seekBackgroundAudio接口，通过e.detail.value将位置信息赋值给position属性也就是播放进度条的位置，以此来控制音乐播放进度。

最后来实现音乐的分享功能，通过使用onShareAppMessage方法，将data数据中的audio.name即当前音乐的歌曲名赋值给title属性，并将音乐播放界面的绝对路径也就是/pages/now/now赋值给path属性，就可以实现从音乐播放界面将当前的音乐分享给其他用户。

至此整个音乐播放器微信小程序就开发完成了，为了使小程序的功能更加完善以及检查是否存在某些问题，在下一章节将会进行音乐播放器小程序的系统检测。

6. 53140122_杨峰_计算机科学与技术_音乐播放器的微信小程序的设计与实现_第6部分 总字数：742

相似文献列表 文字复制比：0%(0) 疑似剽窃观点：(0)

原文内容 红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

第4章音乐播放器的系统测试

4.1测试目的

可以对微信小程序开发系统有一个更加全面地认识，通过对系统中的每一项功能进行测试，不仅可以检测是否完成了上述设计中的每一个模块并查看其运行的情况，是否存在某些问题并进行修正，还可以在这一过程中对现有的功能模块进行完善，使小程序的整体结构更加严禁。

4.2测试及结果分析

表4-1 测试环境：

客户端软件： 微信小程序开发工具

移动端设备： 华为honor9

网络环境： wifi

表4-2 测试范围：

测试项测试内容

页面初始化在打开各个界面的时候整体的布局结构是否与设计的一样

推荐歌曲展示首页中的推荐歌曲模块是否获取到10首音乐

排行榜展示首页中的排行榜模块是否获取到最新的排行榜单

搜索功能在输入关键词之后能否在下方展示出搜索结果列表

歌曲播放在点击某一首歌曲之后能否进行播放

播放控制器在歌曲播放时是否显示出对应的歌曲的封面，歌曲名及歌手

跳转功能在点击歌曲时是否跳转至音乐播放界面

播放界面展示进入播放界面时整体的布局结构是否与设计的一致

封面转动播放界面里的歌曲封面是否跟随音乐的播放而旋转

歌词显示播放界面里歌词是否在对应时间显示

进度条控制歌曲的进度条是否正常显示能够进行拖动。

播放控制点击播放，暂停，下一首按钮时是否正常工作

歌曲信息获取每一首歌曲的封面，歌曲名以及歌手是否获取成功

测试结果分析：本次测试采用的是灰盒测试，在完成所有测试项之后，大部分所有功能都可以正常使用，只是在音乐播放界面中歌词展示部分有时会出现获取不到歌词的情况出现，通过代码跟踪检查，发现获取不到歌词的情况是由于调用接口太频繁而触发的503错误，所以在使用时避免频繁切换歌曲可以避免此问题的发生。

7. 53140122_杨峰_计算机科学与技术_音乐播放器的微信小程序的设计与实现_第7部分 总字数：866

相似文献列表 文字复制比：0%(0) 疑似剽窃观点：(0)

原文内容 红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

第5章总结与展望

5.1总结

本文通过分别介绍微信小程序和音乐播放器来发掘出他们之间的可关联性，从而引出基于微信小程序的音乐播放器的设计与开发。主要的工作如下：

（1）分别阐述了微信小程序和音乐播放器的背景与发展现状，从而展现出本次毕设的课题，音乐播放器的微信小程序的实用性

（2）对微信小程序进行整体框架的分析，从而对微信小程序的开发有了大致的认识。

（3）对将要开发的音乐播放器进行界面及功能设计，并进入实际的开发流程，并在完成开发后对整个系统进行功能测试。

通过对基于微信小程序平台的音乐播放器的开发，不仅让我对微信小程序的整个开发流程有了清晰的认识，也使我对一个完整的软件系统开发有了更深入的理解。回顾整个毕业设计，其实就是对我们在大学四年里收获到的知识的一次整体运用。从对一款软件的需求分析到整体的框架结构设计，再到对其中每一个功能模块的设计与开发，以及最后的测试分析，之前在书本和课堂上学到的知识贯穿于整个开发过程。

5.2展望

本次设计开发的音乐播放器微信小程序中只包含的搜索，推荐和排行榜这几大公共功能，还没有设计针对每个用户的相关功能，如用户的登录，喜欢歌曲的收藏，自定义播放列表等，可以通过在服务器中部署自己的后台系统来实现音乐播放器与后台的通信。同时在界面的样式以及代码的结构方面，还可以做进一步优化。我也会在今后学习工作生活中，不断提高自己，让此项目更加完善。

参考文献

[1]凤凰科技.张小龙分享微信四大价值观将推出“应用号”[DB/OL].
http://tech.ifeng.com/a/20160111/41537673_0.shtml，2016-01-11．
[2]智通编选.微信小程序MAU高达4.7亿，TOP100小游戏占比28%[DB/OL]
https://www.zhitongcaijing.com/content/detail/121235.html，2018-04-25.
[3]陈君,吴冰,王金森.图书馆微信小程序的应用现状及前景分析[J]. 新校园(上旬). 2017(10).
[4]郭毅棋. 基于微信小程序的高校新生预报到系统设计[J]. 厦门城市职业学院学报, 2017, 19(4):10-14.
[5]刘玉佳. 微信“小程序”开发的系统实现及前景分析[J].信息通信，2017,01:260-261.
[6]朱玉强．微信小程序在图书馆移动服务中的应用实践--以排架游戏为例[J].图书馆论坛，2017,37(7):132-138.
[7]王亮, 邹志鹏, 姜虹. 基于微信小程序的医患交流平台的设计与研究[J]. 中国数字医学, 2017(11):71-73.
[8]杨良海, 谭静. 微信小程序衔接自动刷卡结算系统在高校食堂的应用实践[J]. 数字技术与应用, 2017(12):69-69.
[9]高洪涛. 从零开始学微信小程序开发[M].北京:电子工业出版社，2017.
[10]熊普江，谢宇华. 小程序，巧应用:微信小程序开发实战.北京:机械工业出版社，2017

致谢

在吉林大学四年的学习生涯即将结束，回顾这四年的时光，才发现不知不觉中自己已经收获了那么多。在此我要衷心地感谢我的每一位老师，正是他们严谨的教学态度，为我们传道授业解惑，让我在学习的道路上少走了许多弯路。还要感谢大学四年的同窗们，不仅在学习上给予了我许多帮助，还耐心地分享自己的人生经验，让我收获匪浅。

能够顺利的完成本次毕业设计，离不开导师和同学们的帮助。特别感谢刘小华老师对我的关心与支持，在每次我感到困惑的时候，是刘老师和我耐心的沟通交流让我豁然开朗，也使得我少走了许多弯路。再一次向所有的老师表示衷心的感谢！