

基于 Vue 的音乐播放网站的设计与实现

潘晓 黄媛媛

四川大学锦城学院 四川 成都 611731

【摘要】近年来，人们在匆忙的工作或紧张的学习中越来越需要一个能够放松解压的环境，而聆听音乐恰好不失为一个适合各个年龄阶段人群放松的方式。目前市场上的音乐播放平台各式各样，但在最终都离不开商业推广，听音乐的本意就是为了放松，眼花缭乱的广告对用户的体验感大打折扣，所以本音乐播放网站主打简洁、干净的一个音乐播放环境，给用户最舒适的体验感。本文以音乐播放网站为例来介绍怎样利用 Vue 开发框架来设计和实现一个满足用户的音乐欣赏的简洁的个性化音乐平台，通过基于 Vue 开发技术，利用 Node.js 和 MongoDB 数据库就可以实现满足用户需求的应用网站。Node.js+MongoDB 实现后台开发，用于存储用户数据。

【关键词】前端；音乐播放器；Vue 开发框架；Node MongoDB

引言

随着市场上各式各样的音乐播放平台的日趋完善，用户也多了很多选择，那么如何让我们平台在缤纷的音乐播放平台中脱颖而出就成了首要问题。观察发现，目前最火的几个音乐平台或多或少在界面中都掺杂着各式各样的广告，弥漫着浓浓的商业气息，音乐的主要目的是为了让人放松，令人愉悦，所以一个充满广告的音乐平台往往会使得用户体验感大打折扣，由此，一个界面简单，操作简单，不掺杂任何广告的音乐平台就恰好解决了以上问题。本文即通过关键技术选型介绍，功能模块的设计，界面设计分析，具体模块设计，大概介绍本网站的开发流程。

1 技术选型

Vue.js 是一套渐进式的 JavaScript 框架，它只关注视图层，采用自底向上的增量开发设计，目标是通过尽可能简单的 API 实现响应数据绑定和组合的视图组件，采用 MVVM 模式^[1]，此框架开发比较容易，上手快，使用起来较为便捷。因此，将它作为本次网站开发的框架。

Node.js 和 MongoDB 主要用于后台代码编写。配合优势体现在面向文档的，它所能处理的数据格式和前后端 JS 脚本中的数据格式非常接近，这避免了需要在“行”和“数据对象”之间的来回转换。具有高性能、高可行性的优点。MongoDB 可以直接写入数据，不需要提前创建表，方便了代码的书写，查询和插入效率也较高，所以选择这两者搭配来实现后台的创建。

2 功能模块

根据实际需求设计出的功能可以概括为五个部分，

包括用户的登录注册、首页推荐歌单、歌单详情页、歌单排行榜、歌曲搜索、音乐的播放暂停等基础操作功能和歌词文件自动检索并显示功能。

用户登录后能够通过自定义创建歌单，将喜欢的音乐放在创建的歌单中，点击歌单详情页时便可以看到添加的所有喜欢的歌曲，也可以移除喜欢的歌曲。音乐的基础播放暂停等功能可以实现歌曲的暂停、播放、上下曲切换、音量加减、拖拽播放进度条以及播放模式的切换。歌词自动检索显示功能可以在音乐歌词数据库中自动检索当前播放的音乐有无歌词文件，从而进行实时显示歌词或者提示无歌词文件。

3 界面设计

3.1 首页歌单推荐列表

首页分为歌单推荐页、新碟、新歌推荐三个模块，点击歌单可进入歌单详情页，歌单内容包含歌单发布者的头像及昵称，歌单标题及简介，歌单评论，歌单所包含的歌曲，点击歌曲进入歌曲播放页。首页点击歌单广场可进入歌单分类页，分为推荐、精品、欧美、电子等。首页歌单推荐如图 1 所示。



图 1 歌单推荐页

3.2 歌曲播放页及功能按钮

歌曲播放页可以实现播放暂停功能、切换上下曲功能、调节音量大小功能，此外为了方便用户使用体验，还设计了拖拽进度条改变音乐进度条播放的位置功能和改变音乐播放模式功能，单曲循环、列表循环、随机播放。音乐播放的默认播放模式为当前音乐播放列表中循环播放，可通过模式切换功能按钮切换为单曲循环或者随机播放模式。音乐详情内容包含进度滑动条显示播放进度，有标签显示当前的播放时间和总时间信息，显示当前播放音乐的歌名、歌词及歌手信息。如图 2 所示。



图 2 歌曲播放页

3.3 歌曲搜索页

歌曲搜索页分为搜索区、历史搜索区、热搜区及搜索结果区。可输入关键词进行搜索，搜索结果分类，有歌曲、歌手、视频、专辑、歌单、用户等类别。用户搜索后会显示历史搜索结果，也可手动删除搜索结果。热搜显示榜单，根据搜索次数统计。对搜索结果可点击播放，即进入播放页。如图 3 所示。

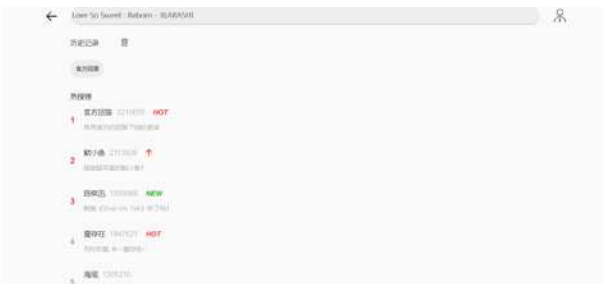


图 3 搜索页

4 后台用户信息

Node+MongoDB 数据库实现的后台部署，MongoDB 创建用户信息表存储个人信息，包含用户 id、用户名及密码。歌单信息表用来存储用户创建的歌单，添加的喜欢的歌曲，包含歌单 id、歌单名、歌曲的一些详细信息。和前端接口连通，用户在登录后可以自行创建歌单，数据传入后台存储创建的歌单信息，可以添加喜欢的歌曲到相应的歌单。具体实现分为三步：第一步后台建立对应的数据信息表。第二步 js 文件连接数据库，建立好对

应的接口，获取用户的输入数据进行判断、存储、读取等操作。第三步前端对应的功能按钮实现功能后和后台接口连通。前后端连通后即可实现以上功能。

4.1 搜索模块

输入功能对输入的关键词进行匹配，显示搜索建议框，每一项分类下都匹配 keyword，显示搜索结果，同时搜索历史也同步更新显示。搜索栏下方显示搜索建议，热搜榜。具体实现：watch 监听 keyword 的长度变化，长度大于 0 时，根据匹配的 keyword 显示搜索建议框，否则隐藏搜索建议框。获取搜索记录，unshift 将关键字插入到数组最前面，存入本地之前进行去重，用函数将搜索的内容放入数组中存入本地，将搜索历史以数组显示。手动点击删除历史记录或者搜索内容改变，对于数组传过来的值，监听变化，对搜索历史内容进行更改。Vue 实现搜索结果页的分类，点击不同的搜索结果分类，根据路由跳转到相应页面。获取历史搜索记录的主要代码如下：

```

getHistory (key) {
    let keys = localStorage.getItem( 'keys' ) ? localStorage.getItem( 'keys' ).split( ',' ) : []
    if (key) {
        keys.unshift(key)
        keys = this.unique(keys)
        localStorage.setItem( 'keys' , keys)
    }
    this.history = keys
    Bus.$emit( 'history' , this.history)
}
    
```

4.2 功能按钮和基本信息模块

首先是实现歌曲的播放和暂停功能按钮，这里首先定义了一个布尔类型变量，初始化为 false，然后在定义的播放函数中对这个布尔值进行判断，为 false 时就通过 toPlay() 来实现播放，否则就利用 toPause() 来实现暂停。然后就是切换上下曲功能，这个要通过初始化的索引值 index 变化来实现，上一曲就通过调用 index-1 可以实现，下一曲就是 index+1 实现。切换播放模式，播放列表中的歌曲放入数组中，顺序播放，则根据歌曲的索引值按顺序播放，随机播放则打乱该数组，再获取随机值播放歌曲，代码 Math.floor(Math.random()*(max-min+1)+min)。

对于音量值的设置通过拖动进度条来改变音量，先设置一个 Slide 进度条，通过获取进度条的值，将其赋值为 setVolum(value) 函数中 value 的值，就可以达到改变音量的效果。最后就是音乐播放进度条显示实时播放进度与通过拖动进度条来改变音乐播放位置，这个就

要同歌词一样先实时获取播放位置，再设置进度条长度值范围，并将获取到的播放位置 position 传给进度条的 value；就可实现进度条随着播放改变随着改变，而通过拖动进度条改变歌曲位置就是相反着的，将拖动到的位置通过赋值给播放的媒体位置就实现了改变的播放位置的功能。拖动播放进度条用的是鼠标事件函数，mousedown、mousemove 和 mouseup。而显示当前播放的时间和时间标签要先通过设置一个函数转换函数，将播放时间转换为分秒形式，就可实时显示时间了^[2]。歌词的时间显示代码如下：

```
let audioTime=minuteValue + '&#x2013;' + second-
Value
this.playTime = audioTime
let barLength = audio.currentTime / audio.duration * 100
this.setProgress(barLength)
const playTime = audio.currentTime + this.offsetLyric
const index = this.getCurrentIndex(playTime, this.rule-
Lyric)
this.nowLyricIndex = index
this.showLyric(index, this.ruleLyric)
this.$refs.lyric.setCurrent(this.nowLyricIndex)
```

4.3 创建歌单及添加喜欢音乐

创建歌单，定义一个用户存放歌单的数组，歌单信息包含歌单 id、歌单名、创建时间及歌单包含的歌曲数。创建歌单时用户需要输入歌单名，点击保存，存储到后台的数据库中。收藏歌单，首先要获取该歌单 id，通过数据库查询操作，找到对应 id 的歌单，返回对应歌单里面的所有内容 data，再添加到用户的歌单收藏列表中，即调用 addPlaylist () 函数，然后 ++this.subscribedCountItem；删除歌单即调用 deletePlaylist () 函数，然后 --this.subscribedCountItem。首先判断该歌曲是否点赞，定义布尔值变量判断 like 等于 1 或 0 来确定，点赞歌曲时默认添加到“我的喜欢”歌单中，同时歌单中的歌曲数 num+1，如果取消点赞，则移除该歌曲，同时 num-1；收藏该歌曲时，弹出提示框，选择收藏到哪一个歌单中，

选择后，根据该歌单 id 查找数据库中的对应歌单，对其进行添加操作，实现歌曲的收藏功能。以下是对歌单的操作，取消收藏此歌单及删除歌单的代码：

```
deleteItem (id) {
  this.drawer = false
  this.$dialog.confirm({
    message: '&#x2013;Are you sure to delete this song
list?&#x2013;
  }).then(_ => {
    if (this.homeFavoritelistSlider) {
      this._cancelPlayList(id)
    } else {
      this._deleteCreatedList(id)
    }
  })
  .catch(_ => {
    this.drawer = false
  })
}
```

5 结束语

本文介绍 Vue 界面设计开发框架的使用，基于此框架的个性化音乐网站已经基本实现，并且可以在浏览器上正常使用。网站实现了歌曲的播放、暂停功能，歌曲的上下首切换、点击列表播放、删除当前播放歌曲自动跳转下一首播放，歌词自动检索显示、播放模式的切换以及拖拽时间和音量的进度条功能。该音乐播放平台实现了个性化、美观、简洁、方便使用的优点，还具有特别好的跨平台特性，具有开发成本低、开发周期短等特点，且代码简单易懂。

【参考文献】

- [1] 易剑波. 基于 MVVM 模式的 WEB 前端框架的研究 [J]. 信息与电脑 (理论版), 2016(19):76-77+84.
- [2] 焦正才, 樊文侠. 基于 Qt/Embedded 的 MP3 音乐播放器的设计与实现 [J]. 电子设计工程, 2012, 20(07):148-150.