

# 基于 Node 和 Vue 的音乐播放系统设计与实现

章 喆 黄媛媛

四川大学锦城学院计算机与软件学院 四川 成都 611731

**【摘要】** 为了给用户提供一款操作便捷、页面美观的音乐播放器，为用户提供良好的音乐体验，设计并开发了一款音乐播放系统，本文简要介绍了这款基于 Vue 和 Node.js 框架开发的音乐播放器网站。网站为了给用户提供轻便简单的音乐播放使用体验，参考了当前各类主流的音乐网站和音乐播放器，构建了一款操作简便、页面美观的音乐播放器网站。本项目主要使用 Vue 框架完成前端界面的制作，使用 Node.js 来搭建服务器，是一款轻便简洁的小型音乐播放网站。用户可以通过使用本网站，体验音乐播放、歌曲推荐、歌单推荐、MV 播放、音乐和歌曲搜索等功能。

**【关键词】** 音乐播放系统; Vue; Node.js; MongoDB

## 0 引言

随着生活水平的提升，音乐已经成为了大多数人必不可少的娱乐需求，而随着信息技术的高速发展，非常多的音乐爱好者都会选择通过网络而不是买光碟来听音乐，现阶段的音乐播放器很多都操作繁琐，广告繁多，为了给用户良好的音乐播放器使用体验，本项目吸收当下主流音乐播放器的优点，并针对各播放器的使用问题，开发了一款操作简洁的音乐播放器。本文提供一种基于 Vue 和 Node.js 框架的音乐播放器的设计思路与实现流程。

## 1 主要技术选型介绍

Vue.js

Vue.js 目前是前端开发领域常用的框架，它是一款可以自顶向下逐层应用的渐进式框架，可以与其他第三方库同时使用，拥有双向数据绑定功能并且鼓励将各模块封装成单独的组件，并通过 API 实现组件的复用。它的这些优点可以减轻开发者的工作量，提高工作效率，非常适合前端开发工作。是一款非常简单、易学习的前端框架。

Node.js

Node.js 是一个搭载了 V8 引擎的 JavaScript 运行环境，可以让 JavaScript 用于开发后端程序，Node.js 通过一同安装的 NPM 包管理工具，使开发者可以下载 NPM 服务器上已有的第三方包、命令行指令，提高了开发者的工作效率，解决了 Node.js 代码部署的问题。

## 2 需求分析

使用音乐播放网站来听音乐的用户可能来自各个年龄层次，因此网站的操作界面应当尽量简洁明了，不宜设置过多跳转链接，将功能尽可能一目了然的展示给用户。音乐播放网站的功能需求大致可以分为用户登录/注册、歌单详情展示、歌单推荐、音乐播放、音乐搜索、歌单搜索、MV 推荐、MV 播放、MV 搜索、歌词展示、评论展示、音乐收藏等功能。使网站能够实现向用户推荐当前热门的歌曲、歌单和 MV，并且用户可以根据自己的喜好查找想要浏览的音乐、歌单和 MV 资源。

## 3 开发模式可行性分析

本项目采用前后端分离模式开发，使用 Vue 框架构建页面，并利用 Element-UI 进行页面布局设置，后端部分使用 MongoDB 数据库，对数据进行存储，利用 Node.js 框架运行程序，搭建服务器。本项目通过使用前后端分离的开发模式，提高工作效率，使前后端可以同时进行开发，提高开发效率，使前后端的修改不会相互影响，如果后端数据出错，前端页面也可以照常访问，但无法显示数据。通过对路由的配置，实现了网站资源的按需加载，提升用户的交互体验。项目开发的可行性很高。

## 4 功能模块设计

网站根据功能分类将页面分为发现音乐、推荐歌单、最新音乐、最新 MV、音乐盒、登录/注册等页面进行开发，其中发现音乐部分作为主界面向用户展示网站所拥有的大多数基本功能，向用户推广当前热门的音乐、歌单、MV，其他页面则负责展示网站所拥有的总体数据，为用户提供登录注册功能。模块设计如图 1 所示

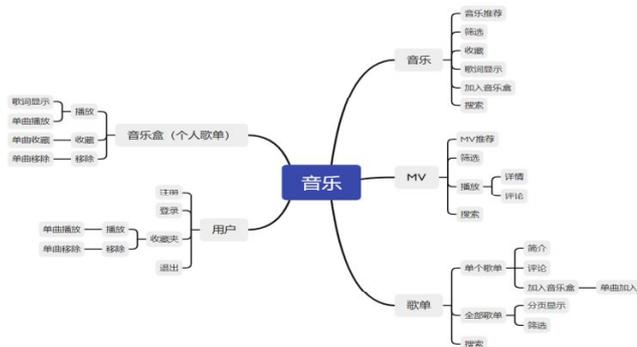


图 1 功能模块设计

## 5 功能描述

用户注册、登录:

注册界面用于为未注册用户注册功能，注册流程如下:

用户在注册界面填写用户名、邮箱、密码和确认密码数据

获取用户输入数据，进行数据校验，通过则继续，否则提示出错;

发送 ajax 消息到后端;

后端收到消息, 将用户信息添加到数据中, 返回登录界面;

本系统中用户可以通过用户名、密码进行登录, 登录流程如下:

用户在登录界面填写邮箱和密码。

获取用户输入的邮箱和密码数据, 校验数据, 符合规范则继续进行下一步操作, 否则提示出错。

发送 ajax 消息到后端;

后端收到消息, 比较数据库中存储数据, 如果存在相应数据, 返回登录成功, 页面跳转进入首页。

发现音乐:

发现音乐界面由顶部轮播图、推荐歌单、最新音乐、推荐 MV 四个部分构成。用于向用户推荐网站目前最新的热门歌曲、歌单和 MV。是用于展示网站的主要功能的重要页面, 是音乐播放系统的主界面。

轮播图用于展示广告或音乐宣传, 向用户提供网站热门信息。图片从数据库导入, 定期更换。

推荐歌单部分用于展示当前网站热门的歌单, 点击歌单图标会跳转进入歌单详情界面, 将选中的歌单的数据传递给歌单详情界面, 向用户显示选中歌单的详细内容。歌单推荐组件展示歌单的封面和名称, 向用户介绍歌单的一些基本信息。

最新音乐部分用于推荐当前热门的音乐, 点击图标选择音乐即可播放音乐。音乐单曲组件会展示音乐的封面、音乐的名字和歌手等信息, 可以使用户方便的了解到歌曲的基本信息。

推荐 MV 部分用于介绍热门 MV, 点击 MV 进入 MV 详情界面, 将 MV 数据传递给 MV 详情组件, 并播放 MV, 显示 MV 的详细信息。MV 推荐组件向用户展示 MV 的播放量、MV 封面、MV 名称和 MV 的歌手等信息, 便于用户选择。

关键代码如下:

访问 API 样例:

```
axios({
  url: «https://autumnfish.cn/personalized»,
  method: «get»,
  params: {
    limit: 10,
  },
}).then((res) => {
  console.log(res);
  this.list = res.data.result;
});
```

推荐歌单:

用于展示网站拥有的歌单, 用户可以通过点击歌单图标进入歌单详情界面浏览歌单内容。可以通过歌单的标签实现筛选, 用户点击标签即可筛选得到拥有相应标签的歌单。歌单简介用于介绍歌单发布人对歌单的简要介绍, 歌单评论区显示其他用户对歌单的评价, 评论顺序根据时间先后排列。

最新音乐:

根据音乐发布时间显示最新音乐, 可以根据语言对

音乐进行分类, 点击音乐即可播放。页面不设置分页, 随用户浏览加载

最新 MV:

展示网站所拥有的 MV, 用户可点击 MV 进入 MV 详情界面播放 MV。点击 MV, 通过接口将 MV 信息传递给 MV 详情界面, 在 MV 详情界面播放选择的 MV。本页面提供地区和类型两种属性, 用户可以通过选择属性筛选所需要的 MV 类别。本页面同时具有排序功能, 用户可以通过选择想要的排序方法浏览喜欢的 MV。

音乐盒:

本功能仅供登录用户使用, 用户可以在播放器播放选择收藏功能, 将喜欢的歌曲加入音乐盒。

点击播放器组件上的收藏按钮, 将当前播放的音乐数据传入音乐盒创建函数中, 在音乐盒内添加一个拥有当前音乐数据的音乐组件。

搜索功能:

用户可通过在搜索框输入关键字搜索需要查找的歌曲或 MV

用户输入回车键或点击搜索按钮后, 比对拥有相应关键词的歌曲、歌单、MV, 并将搜索结果展示, 用户可以通过点击歌曲、歌单、MV 的分类选择查找的内容。

歌单详情:

歌单详情页面用于展示歌单的详细内容, 包括歌单的封面、发布用户、标签、简介、歌曲列表、评论和播放全部功能。

加载该界面时获取储存的歌单数据, 展示歌单详情功能。点击播放全部按钮, 立即从歌单的第一首歌曲开始播放歌单歌曲。

评论功能展示其他用户对当前歌单的评价, 一页显示 10 条评论, 一次显示 5 个分页标签。

## 6 页面设计

本网站在所有页面中的页面布局, 主要由顶部导航栏、页面主体、播放器、歌词显示、回到顶部、底部区域部分构成。具体布局和功能设计如下:

导航栏通过路由跳转实现不同功能页面的切换, 并改变当前页面的切换按钮的颜色。导航栏颜色为黑色, 文字为白色, 选中状态主题为黄色。

主体部分是用于放置网站主要内容的部分, 将页面作为一个整体封装成组件, 通过改变主体部分展示的组件完成页面切换。

播放器用于实现音乐的播放, 在点击选择的歌曲后将从接口中获取的歌曲信息传递给播放器组件, 实现歌曲播放。

选择歌曲时将歌词数据传递给显示歌词部分, 歌词显示组件在页面加载时被隐藏, 点击显示歌词按钮显示被隐藏的歌词显示部分组件, 再次点击会再次隐藏歌词显示组件。

显示歌词、回到顶部和播放器部分都相对浏览器页面固定, 不会移动。

## 7 数据库设计

数据库采用 MongoDB 数据库进行搭建。命名英文字母加自然数组成, 用于存储用户信息和用户音乐盒数据, 包含用户信息表和音乐盒数据表, 其中用户信息表要求

name 属性和 email 属性要求不能重复, 用户信息表和音乐盒表的 box 属性关联, 方便维护时对音乐盒表数据的管理。

## 8 结束语

本音乐播放网站主要采用 Vue.js 和 Node.js 框架开发, 网站可以为用户提供音乐播放、MV 播放、音乐搜索、歌单推荐、歌单搜索、MV 搜索、歌单播放等功能, 经过测试验证, 目前本网站已经基本实现了用户对音乐播放网站的需求。目前音乐盒功能和评论功能仍需完善, 会在后续的开发中对其中存在的一些问题进行修复。

## 【参考文献】

[1] 王金龙, 宋斌, 丁锐. Node.js: 一种新的 Web 应用构建技术 [J]. 现代电子技术, 2015, 38(06): 70-73.

[2] 徐岷, 朱广华, 贾瑶. 基于 VueJs 的 WEB 前端开发研究 [J]. 科技风, 2017(14): 69.