

基于 Vue 的音乐平台前端设计与实现

洪晨馨 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】为了能够更简便地播放音乐,设计实现了一个操作简单、页面简洁的音乐平台。本文主要简述了基于 Vue 框架设计开发的音乐平台的前端页面实现流程。此项目主要采用 Vue 框架来搭建页面框架,使用 MongoDB 实现用户注册登陆数据存储,引入 Element UI 做样式辅助。在本系统中,用户可以播放音乐,通过 MV 观赏对应音乐视频,搜索想听的音乐,注册登陆后将喜欢的音乐加入音乐盒。

【关键词】音乐播放器; Vue; MongoDB

1 引言

随着社会发展,人们不仅只追求物质层面上的满足,精神上的需求也日益渐增,听音乐已经成为各年龄段人们都会做的事情,很多人也只需要一个能够欣赏音乐而没有其他信息干扰的简洁音乐播放平台,因此,设计一个页面干净、操作方便的音乐平台极有意义。本平台操作简单,页面简洁干净,提供了播放音乐、播放 MV、注册登录、收藏、推荐等功能。本文给出基于 Vue,辅以 Element UI、MongoDB 技术的音乐播放器前端页面的实现流程。

2 技术综述

2.1 Vue.js

Vue.js 是构建用户界面的渐进式框架,以数据驱动的组件化系统为核心思想。组件化是指,把用户界面拆分成一系列相互嵌套的组件,每个组件相互独立,具有相互独立的业务逻辑,又能通过数据传递和其他组件相互联系^[1]。Vue.js 让开发者不修改 DOM,只关注事件逻辑,用户在页面操作后,通过数据改变直接让页面重新渲染 UI。

2.3 MongoDB

采用文档型数据库 MongoDB 存储数据,主要原因是 MongoDB 在保证海量数据存储的同时,还具有高效率的查询性能^[2]。它查询快速,操作简单。MongoDB 的写操作会被自动物理写到日志中 100 微秒内,写操作此时就变成了持久了的,因为在事故发生后启动,在数据恢复时,数据仍然可被恢复^[3]。

3 系统分析与设计

3.1 需求分析

本平台拟实现如下功能:

(1) 歌曲推荐和播放: 首页显示最新歌曲列表,点击歌曲封面播放按钮或双击歌曲名称实现歌曲播放。

(2) 歌单推荐和播放: 首页显示推荐歌单列表,单击歌单封面进入单个歌单详细页面并播放歌单包含歌曲。同时歌单页还有歌单简介和用户评论功能。

(3) MV 推荐和播放: 首页推荐 MV 列表,单击 MV 进入对应 MV 详细页面并播放此 MV,且页面内还可以查看、

播放推荐的类似 MV,下面也附带有用户评论功能。

(4) 用户注册、登录、退出: 用户点击顶部菜单栏的登录按钮进入登陆页面,若无账号,则点击注册按钮进行注册。登录后的用户可在账号下拉列表框内选择退出登录。

(5) 音乐盒的歌曲收藏、移除: 登录的用户遇到喜欢的歌曲可以点击播放框内的灰色爱心图标将其加入音乐盒,且爱心显示红色,放入音乐盒内的歌曲可以进行单曲删除或全部移除操作。被移除音乐盒的歌曲爱心会改变为灰色。

(6) 搜索: 用户在顶部菜单栏输入框输入想搜索的内容,按下回车键开始搜索,若输入内容为空,跳出错误提示;若合法,进入对应搜索页面,页面内可查看对应歌曲、歌单和 MV。

(7) 回到顶部: 若用户在页面内下页面,可点击回到顶部迅速回到页面顶部。

3.2 模块设计

本项目由音乐、MV、歌单、用户、音乐盒、返回顶部六个模块组成。

其中,音乐模块负责对歌曲的推荐、筛选、收藏、搜索和播放,是页面的所有歌曲操作的管理模块也是最核心的部分,包括最基本的功能;MV 模块则是对 MV 进行推荐、筛选、播放和搜索,负责操作 MV 部分功能,同时允许查看用户对此 MV 的评论;歌单模块用于推荐歌单、筛选、播放歌单内所含歌曲,同时允许用户查看对应歌单的用户评论;用户模块有注册登录、退出、使用音乐盒模块功能的作用;音乐盒模块主要能够播放加入音乐盒的音乐以及移除音乐;返回顶部模块则是为了方便用户在滑动到页面下方时能快速回到顶部。用户模块与音乐盒模块有相互关联性,音乐模块的收藏功能也与用户模块相关联。模块设计图如图 1 所示。

3.3 数据库设计

数据库采用的是 MongoDB 文档型数据库。本平台使用了一个数据表: 用户信息表(users)来存储用户信息。其中包括了编号(_id)、姓名(name)、邮箱(email)、密码(password)以及日期(date)五个值。



图 1 模块设计

4 系统模块

4.1 系统功能模块设计

4.1.1 音乐模块

对于音乐模块的页面，主要有两块功能：一块是切换的分类标签，一块是显示的音乐数据。

首先切换分类标签：使用 watch 实现数据监听，侦听分类 tag 的改变来切换每个分类页，此处对应筛选功能。

其次显示数据：

- (1) 用户点击歌曲按钮；
- (2) 页面使用 axios 后台调取外部数据接口；
- (3) 获取的数据传回前端渲染到页面；
- (4) 数据显示。

此处歌曲时长显示做了一点处理，接口获取到的数据是毫秒数，因此通过除法转化为分秒。

对于音乐播放功能：

用户点击音乐封面或双击歌曲名称；
页面使用 axios 后台调取外部数据接口；
获取到对应 url 数据传回前端页面；
音乐播放。

对于搜索功能页面，分为两个显示部分：第一部分是标题，即所搜索的内容，后面跟上搜索出来的结果数据有多少条，标题随着输入搜索框内容不同动态改变。下一部分有三个切换标签：歌曲、歌单和 MV。

歌曲标签下是一个大表格，存放标题、歌手、专辑和时长四栏，设计思路类似最新音乐，显示 10 条数据后使用 element-ui 的分页器来分页显示；、

歌单和 MV 标签页面类似，使用一个大 div 存放所有内容，每个歌单、mv 都由接收的数据动态渲染在一个小 div 内，歌单页显示 5*2 条数据，MV 页面显示 4*3 条数据，剩下的数据分页显示。

对于搜索功能，数据的显示和 tag 页的切换同音乐页面。只是多了对于 MV 时长显示以及 MV、歌单播放量的数据显示处理。

用户在顶部搜索框输入想查找的内容；

页面获取搜索内容；

若搜索内容为空，返回提示；

若搜索内容不为空，页面使用 axios 后台调取外部数据接口；

将搜索内容与数据匹配，若对应匹配，返回对应数据渲染到页面。

4.1.2 MV 模块

MV 模块页面搭建思路、数据传递、筛选、搜索功能类似音乐模块，此处不做赘述。新增了引用 element-ui 实现的分页功能，以及单击单个 MV 后跳转到详细播放页面的功能：使用 \$router.push() 实现跳转。每个 MV 上做了时间显示处理，将获取到的毫秒数转化为分秒，方便用户查看。

跳转到的详情页页面分为两大块：

顶部一块为 MV 显示区域以及相关 MV 推荐；下部是评论区，两部分的数据都由 axios 调用外部接口后将获取的数据传输到空数组内，再由数组内使用 v-for 循环遍历分别显示到对应区域。

4.1.3 歌单模块

歌单模块实现思路中和了音乐与 MV 模块，搜索、筛选、分页等通用功能实现方式一致。单个歌单单击通过 \$router.push() 进入详情页面，页面分为上下两块：顶部歌单简介，下部为歌单内包含的歌曲，两部分皆事先搭好页面框架样式，再由接口调入数据进行渲染。歌曲时间也对应做了时间转化处理。

4.1.4 用户模块

对于新用户，在不注册的情况下无法登录，注册界面中用户需要输入用户名、邮箱、密码和确认密码。用户名不能为空且长度在 2 到 30 个字符之间；密码不能为空且长度在 6 到 30 个字符之间；邮箱设置格式为邮箱类型，输入首先要符合格式，且后台会查询数据库，若邮箱已被注册，则注册失败；确认密码首先和密码限制条件一直，另新增了一个限制：与密码值一致。注册使用了 bcrypt 加密工具，调用 bcrypt.genSalt() 和 bcrypt.hash() 进行密码的加密。

注册后的用户跳转到登录页面，登陆前提是邮箱格式正确且密码不为空，输入邮箱时会进行数据库查询，搜索用户是否正确，不存在则返回 404 状态码。此处也使用了 bcrypt 工具，调用 bcrypt.compare() 实现输入密码与注册密码的匹配。

4.1.5 音乐盒模块

音乐盒模块主要是歌曲的收藏功能，以及对收藏歌曲的移除。

使用 \$store.commit() 的 add 和 delete 来添加和删除音乐

通过 \$store.getters.musicId()、\$store.getters.user() 来监听获取歌曲 id 和用户；

加入收藏的歌曲使用一个判断语句来改变类名，以此判断是否为喜欢的歌曲。

同时使用红色爱心图标对应加入音乐盒的音乐，灰色爱心图标代表未加入音乐盒，以方便用户查看。

4.1.6 返回顶部模块

此模块相当于一个小功能，主要通过 requestAnimationFrame() 和 cancelAnimationFrame() 函数对动画的改变来实现返回顶部效果。

5 结束语

本平台使用了 Vue 框架和 MongoDB 技术来进行开发实现,文中简述了整个实现流程,通过 Vue 搭建页面框架,Element UI 补充样式,使用 MongoDB 进行用户注册登录的数据存储,使用 axios 调用外部接口数据实现数据传递。目前实现的页面简洁美观,满足用户需求,但是还存在一些需要完善的地方,在往后的更新中会逐步解决各种需要改善的问题,以求成为功能完整、用户体验感好的简洁音乐播放平台。

【参考文献】

[1] 黄佛辉. 基于 Vue.js 的 WebGIS 开发研究 [D]. 重庆交通大学, 2017:

[2] 吕林. 基于 MongoDB 的应用平台的研究与实现 [D]. 北京邮电大学, 2015:

[3] 曾明宇. 一种基于 Storm 和 Mongoddb 的分布式实时日志数据存储与处理系统的设计与实现及应用 [D]. 浙江大学, 2015: