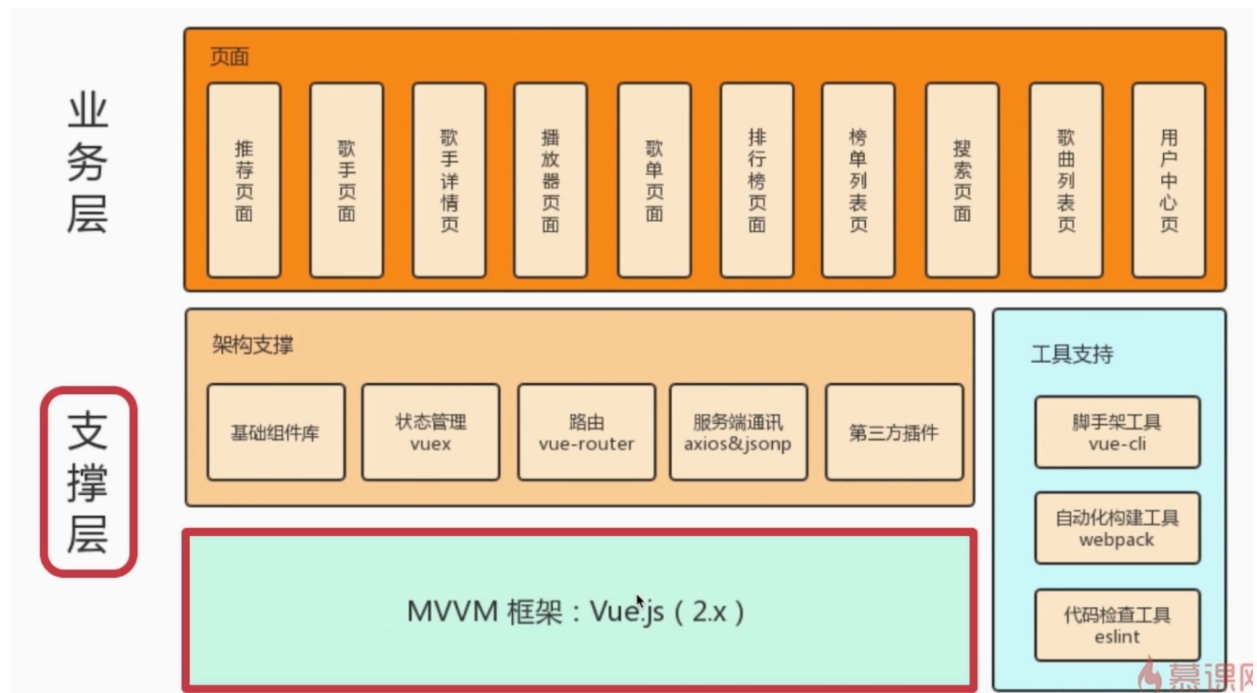


## 项目概述



## 课程安排

- 第 1 章课程介绍
- 第 2-12 章全程实战开发
- 第 13 章项目优化及编译打包
- 第 14 章回顾总结

## 技术要点

- global
  - config folder

- common
- base component
- api
- fastclick
- router
- store
- js-base64
- jsonp
- lazyload in vue
- better-scroll
- create-keyframe-animation
- router
- vuex
- music player
  - audio
  - [SVG circle](#)
  - lyric-parser
- demo feature
  - 微信好友列表
  - 上拉列表标题不透明度
  - touch滑动切换

## 模块说明

api.server.js - api代理转发

QQ音乐部分api对域（http request - host）进行了权限的限制（403 - Forbidden）；

前端部分，也就是axios不能设置http请求头，只能传参；

这里用express搭建一个转发代理服务，把localhost:8080/api的请求转发到

localhost:300/api，在该代理服务向QQ音乐的api发送请求。

该服务可单独使用也可作为以模块在vue-cli中导入使用。

## api - api模块

config.js 配置了api的一些通用参数;

同级目录下的其他文件都是api模块, 供组件发送http请求, 响应数据;

使用axios HTTP库, 发送http请求到QQ音乐的api (部分api通过代理转发 - api.server.js), 返回一个promise对象

编写api模块时, 应确认该请求是否需要传参, 错误处理等

## base - 基础组件库

一些可公共的组件, 可单独抽象出整合到这里;

如listview, loading animation, progress-bar/circle, scroll, silder等

抽象出基础组件时, 基础组件的数据必须为外部传入, 以及组件初始化 (无数据), 组件事件传递, 组件页面功能应编写在基础组件, 自定义功能应向外部抛出。

## common - 字体 图片 样式 基础js库

fonts采用icon fonts生成, 作为模块在css预处理语言中导入使用;

stylus也分为几个模块, minxin reset icon base 和index模块入口;

js给所有的组件/router/store调用, 包含一些配置参数, dom, jsonp以及数据处理的一些模块, singer.js和song.js都返回一个类, 整合了数据格式。

## components - 页面布局组件

项目的每个页面, 组件都从这里引用;

布局使用基础组件库和第三方插件构成,

发送http请求, 与vuex绑定, 渲染数据;

实现用户交互功能与路由挂载;

## router - 单页面应用页面管理器

## store - 状态管理, 共享数据

vuex应用讲模块划分为state action mutation, 这是最基本的模块;

musicplayer项目新增了几个模块;

- mutation-type 定义mutation动作的函数名, 以便在不同场景下使用;
- getters获取store中state仓库的值, 也可派生状态

App.vue - 全局组件

全局组件作为容器使用，只导入布局组件，挂载router组件

main.js - 项目入口文件

- 导入全局样式
- 导入插件
- 初始化vue项目

#1 bug 点击播放列表，播放功能出现store错误

```
✖ Error: [vuex] Do not mutate vuex store state outside mutation handlers. vue.runtime.esm.js?ff9b:1737
  at assert (vuex.esm.js?358c:97)
  at Vue.store._vm.$watch.deep (vuex.esm.js?358c:746)
  at Watcher.run (vue.runtime.esm.js?ff9b:3229)
  at Watcher.update (vue.runtime.esm.js?ff9b:3203)
  at Dep.notify (vue.runtime.esm.js?ff9b:693)
  at Array.mutator (vue.runtime.esm.js?ff9b:840)
  at eval (singer-detail.vue?1be9:52)
✖ [Vue warn]: Error in callback for watcher "function () { return this._data.$$state }": "Error: [vuex] Do not mutate vuex store state outside mutation handlers." vue.runtime.esm.js?ff9b:587
(found in <Root>)
```

```
▶ mutation SET_SEQUENCCEL_IST @ 13:07:33.053
▶ mutation SET_PLAY_LIST @ 13:07:33.056
▶ mutation SET_CURRENT_INDEX @ 13:07:33.059
▶ mutation SET_PLAYING_STATE @ 13:07:33.062
▶ mutation SET_FULL_SCREEN @ 13:07:33.066
```

技巧

定义变量 data computed watch属性中定义的数据，是响应式数据  $V \rightleftharpoons M$ ，不需要与页面交互，只是作为js逻辑交互的控制变量，可在生命周期函数created中将该属性挂载到this实例

善用if return 控制流程

Math.max Math.abs 是两个好东西

```
middleTouchStart(e) {
  this.touch.initiated = true
  const touch = e.touches[0]
```

```

        this.touch.startX = touch.pageX
        this.touch.startY = touch.pageY
    },
    middleTouchMove(e) {
        if (!this.touch.initiated) {
            return
        }
        const touch = e.touches[0]
        const delatX = touch.pageX - this.touch.startX
        const delatY = touch.pageY - this.touch.startY
        if (Math.abs(delatY) > Math.abs(delatX)) {
            return
        }
        const left = this.currentShow === 'cd' ? 0 : -window.innerWidth
        const offsetWidth = Math.min(Math.max(-window.innerWidth, left + delatX),
0)

        this.touch.percent = Math.abs(offsetWidth / window.innerWidth)
        this.$refs.lyricList.$el.style[transform] =
`translate3d(${offsetWidth}px, 0, 0)`
        this.$refs.lyricList.$el.style[transitionDuration] = ''
        this.$refs.middleL.style.opacity = 1 - this.touch.percent
        this.$refs.middleL.style[transitionDuration] = ''
    },
    middleTouchEnd() {
        let offsetWidth, opacity
        if (this.currentShow === 'cd') {
            if (this.touch.percent > 0.1) {
                offsetWidth = -window.innerWidth
                opacity = 0
                this.currentShow = 'lyric'
            } else {
                offsetWidth = 0
                opacity = 1
            }
        } else {
            offsetWidth = 0
            opacity = 1
        }
    }
}

```

```
    if (this.touch.percent < 0.9) {
      offsetWidth = 0
      this.currentShow = 'cd'
      opacity = 1
    } else {
      offsetWidth = -window.innerWidth
      opacity = 0
    }
  }

  const time = 300
  this.$refs.lyricList.$el.style[transform] =
`translate3d(${offsetWidth}px, 0, 0)`
  this.$refs.lyricList.$el.style[transitionDuration] = `${time}ms`
  this.$refs.middleL.style.opacity = opacity
  this.$refs.middleL.style[transitionDuration] = `${time}ms`
}
```

记录touch.pageX touch.pageY

减去touch.startX touch.startY

**touchMove**

获得交互移动距离

纵向滚动，不执行任何操作

**touchEnd**

middleL 使用translate指定位移尺寸

前置补零

---

