**项目说明/初稿**

**本设计分为三部分（页面）：**

**我的**

**作品**

**动态**

**打包时注意：**

项目打包前需要将main.js的路由mode从history改为hash;

**启动方式（必须同时启动）：**

window + R，输入cmd，并进入文件的对应位置（以下两个文件的位置）

catamusic 启动方式：npm run serve

NeteaseCloudMusicApi 启动方式：node app.js

**图片懒加载**

安装：npm install vue-lazyload --save-dev

导入： import VueLazyload from 'vue-lazyload'

Vue.use(VueLazyload)

使用：<img v-lazy="">

入口页面 index.html、App.vue

程序入口文件 main.js(加载路由、组件、初始化Vue实例等)

**路由(routes ./src/routes)：**

**index.js 链接以下路由(【】内代表它们在index.js中导入后的名称，仅routes下的index.js)**

1. **Home 链接 主页/我的 视图 【HomeRouter】**
2. index.js (Home路由/主页路由)
3. login.js (登录路由) 【Login】
4. userDetail.js (用户详情路由) 【UserDetail】
5. playlist.js (歌单详情) 【PlayList】
6. **Rec 链接 推荐 视图 【RecRouter】**
7. index.js (Rec路由/推荐路由)
8. **Dynamic 链接 动态 视图 【Dynamic】**
9. index.js (Dynamic/动态路由)
10. **Search 链接 搜索 视图 【Search】**
11. index.js (Search路由/搜索路由)
12. searchDetail.js (Search路由/搜索详情路由) 【SearchDetail】
13. **Record 链接 记录 视图 【Record】**
14. index.js (Record/记录路由)

children: /Cloud、/PlayRecord

**视图(views ./src/views):**

1. **index.vue** 引入公共组件，如播放器 **（./src/views）**
2. SwitchBottom
3. Player
4. PlayerChart
5. SearchTop
6. PlayerSong
7. <router-view></router-view
8. **home** 主页 id=“home” **(./src/views/home）**
9. User
10. **rec** 动态 id=“rec” **(./src/views/rec）**
11. Rec
12. **dynamic** 推荐id=“dynamic” **(./src/views/dynamic）**
13. Dynamic
14. **search**  搜索 id=“search” **(./src/views/search)**
15. Search
16. **Record** 记录 id=“record” **(./src/view/record)**

<router-view></router-view

**组件(components ./src/components，【】内为引入view时的名称)：**

1. **decorate**  存放一些装饰
2. 顶部搜索框 id=“searchTop” 【SearchTop】
3. 底部选项卡 id=“switchBottom” 【SwitchBottom】
4. 播放音乐 id=“songAudio” 【songAudio】
5. **player** 存放播放器
6. 播放器 id=“player” 【Player】
7. 播放列表 id=“playerChart” 【PlayerChart】
8. 播放界面 id=“playerSong” 【PlayerSong】
9. 歌单 id=“playerList” 【PlayerList】
10. **playerPublic.js 用于跨页面传递/调用方法**
11. **search** 存放搜索页面及搜索的内容
12. 搜索页面 id=“search” 【Search】
13. 搜索内容 id=“searchDetail” 【SearchDetail】
14. **home** 存放主页的组件
15. 用户信息 id=“user” 【User】
16. 用户登录 id=“login” 【Login】
17. 用户详情 id=“userDetail” 【UserDetail】
18. **rec** 存放推荐的组件

rec.vue

1. 歌单推荐 id=“resourse” 【Resourse】
2. 新曲推荐 id=“songs” 【Songs】

**vuex (./src/vuex)**

store.js //存储状态

* 1. audio: '',//存放歌曲的src地址
  2. mode: 0,//用于歌曲播放状态的切换,默认顺序播放(0:顺序播放，1:随机播放，2:单曲循环)
  3. InputValue: '',//搜索的关键字
  4. songList: [],//存储单首音乐的所有属性
  5. songsList: [],//存储多首音乐的所有属性
  6. songsLength: '',//存储多首音乐的长度
  7. topList: [],//存储排行榜
  8. topId: '',//存储歌单id

**$route.meta**

* 1. detail 控制主页的顶部搜索框的显示（#searchTop，true为不显示，目前使用的有：routers下的Home/userDetail、Search/searchDetail.js、Home/playlist.js、Record/index.js）
  2. bottom 控制播放器的位置（#player，botton: 0, 目前使用的有：routers下的Home/userDetail、Search/index.js、Search/searchDetail.js、Home/playlist.js、Record/index.js）
  3. istrun 控制底部选项卡的显示（#switchBottom，false为不显示,目前使用的有：routers下的Dynamic/index.js、Home/index.js、Rec/index.js、Search/index.js、Search/searchDetail.js、Record/index.js）
  4. login 控制播放器是否显示（#player，true为不显示, 目前使用的有：router下的Home/login.js）

**axios (plugins ./src/plugins)**

用于连接网易云API，文档地址：https://neteasecloudmusicapi.vercel.app