

前端菁英班大作业介绍

网易云音乐

目录

目录

-  项目概览

目录

-  项目概览
-  功能介绍

项目概览

使用的技术

项目概览

使用的技术

- React - 使用React作为前端框架，以组件为单位，代码更加模块化

项目概览

使用的技术

- React - 使用React作为前端框架，以组件为单位，代码更加模块化
- Axios - 使用Axios与后端进行通信，配置简单、扩展性强

项目概览

使用的技术

- React - 使用React作为前端框架，以组件为单位，代码更加模块化
- Axios - 使用Axios与后端进行通信，配置简单、扩展性强
- Material-UI - 使用Material-UI作为CSS框架，全局采用Material风格，简洁美观

项目概览

使用的技术

- React - 使用React作为前端框架，以组件为单位，代码更加模块化
- Axios - 使用Axios与后端进行通信，配置简单、扩展性强
- Material-UI - 使用Material-UI作为CSS框架，全局采用Material风格，简洁美观
- TypeScript - 采用TS编写，增加了代码的可读性和可维护性

项目概览

使用的技术

- React - 使用React作为前端框架，以组件为单位，代码更加模块化
- Axios - 使用Axios与后端进行通信，配置简单、扩展性强
- Material-UI - 使用Material-UI作为CSS框架，全局采用Material风格，简洁美观
- TypeScript - 采用TS编写，增加了代码的可读性和可维护性
- React-Router - 使用React-Router作为路由库，实现了代码缓冲加载，页面加载速度更快

项目概览

使用的技术

- React - 使用React作为前端框架，以组件为单位，代码更加模块化
- Axios - 使用Axios与后端进行通信，配置简单、扩展性强
- Material-UI - 使用Material-UI作为CSS框架，全局采用Material风格，简洁美观
- TypeScript - 采用TS编写，增加了代码的可读性和可维护性
- React-Router - 使用React-Router作为路由库，实现了代码缓冲加载，页面加载速度更快
- GitHub Actions - 持续集成

项目概览

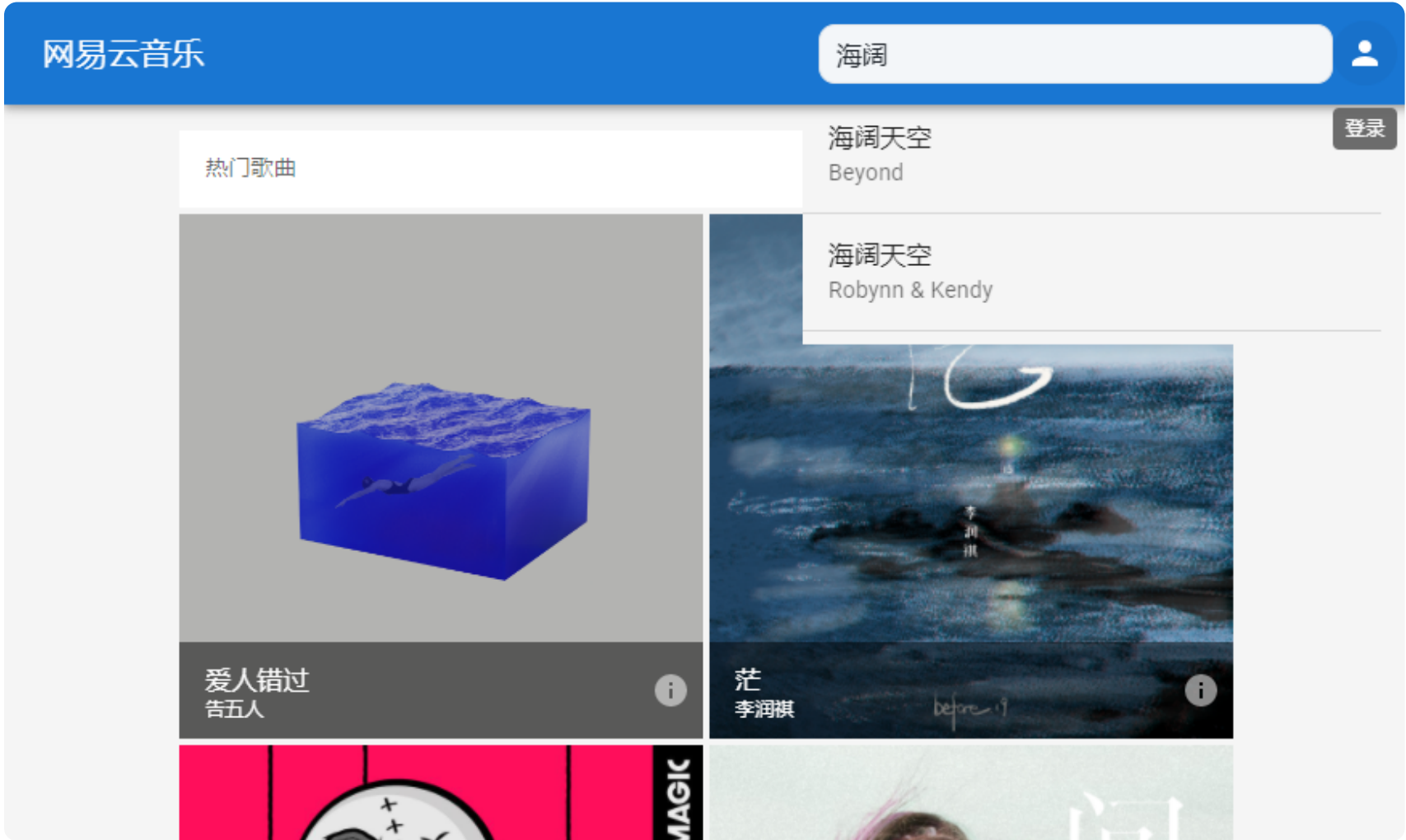
使用的技术

- React - 使用React作为前端框架，以组件为单位，代码更加模块化
- Axios - 使用Axios与后端进行通信，配置简单、扩展性强
- Material-UI - 使用Material-UI作为CSS框架，全局采用Material风格，简洁美观
- TypeScript - 采用TS编写，增加了代码的可读性和可维护性
- React-Router - 使用React-Router作为路由库，实现了代码缓冲加载，页面加载速度更快
- GitHub Actions - 持续集成
- Git Hooks - 提交的时候检查并格式化代码

功能介绍

整体框架

网页由两部分组成：顶部导航栏和页面窗体

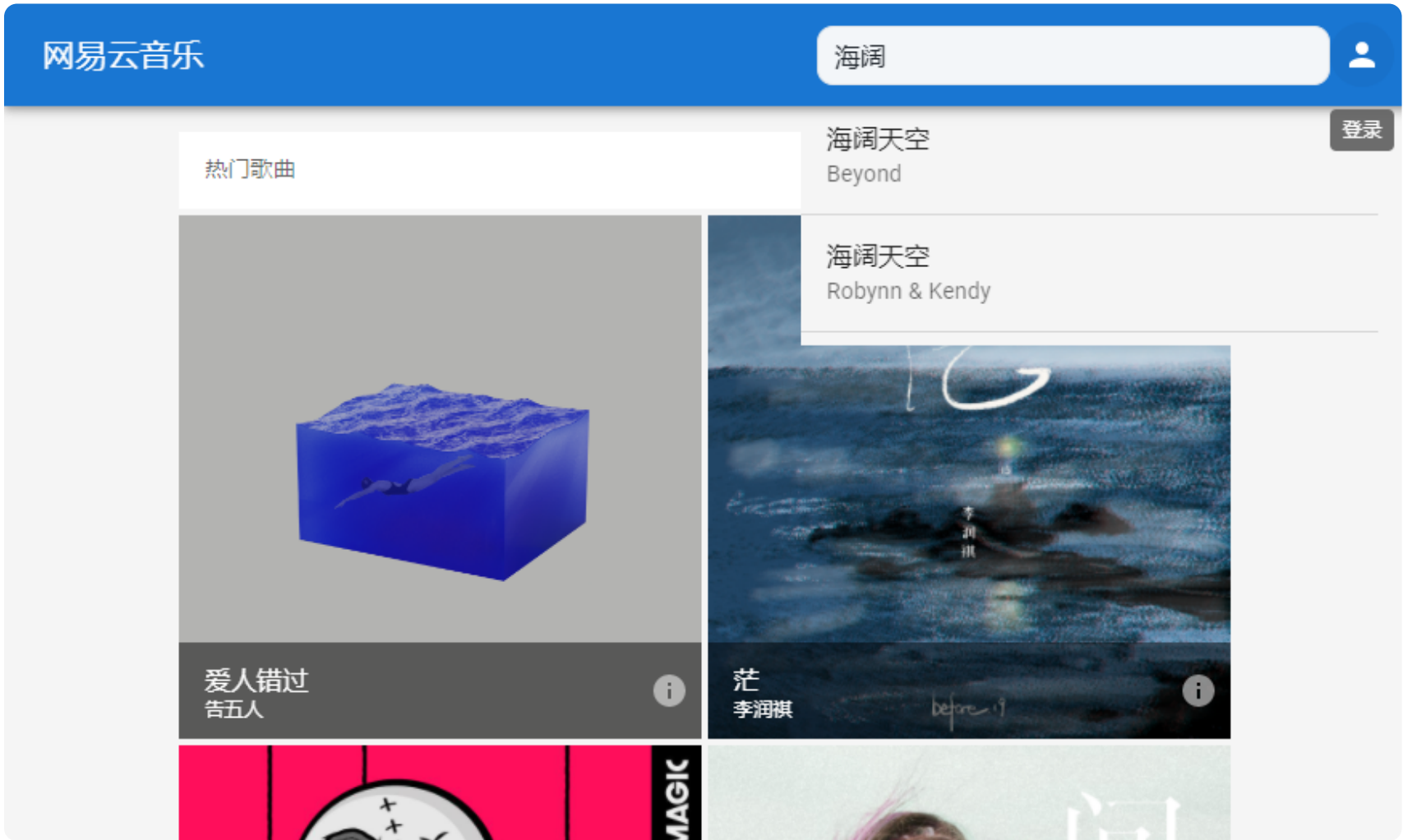


功能介绍

整体框架

网页由两部分组成：顶部导航栏和页面窗体

- 顶部导航栏: 搜索 + 用户状态
 - 搜索栏
 - 用户状态信息

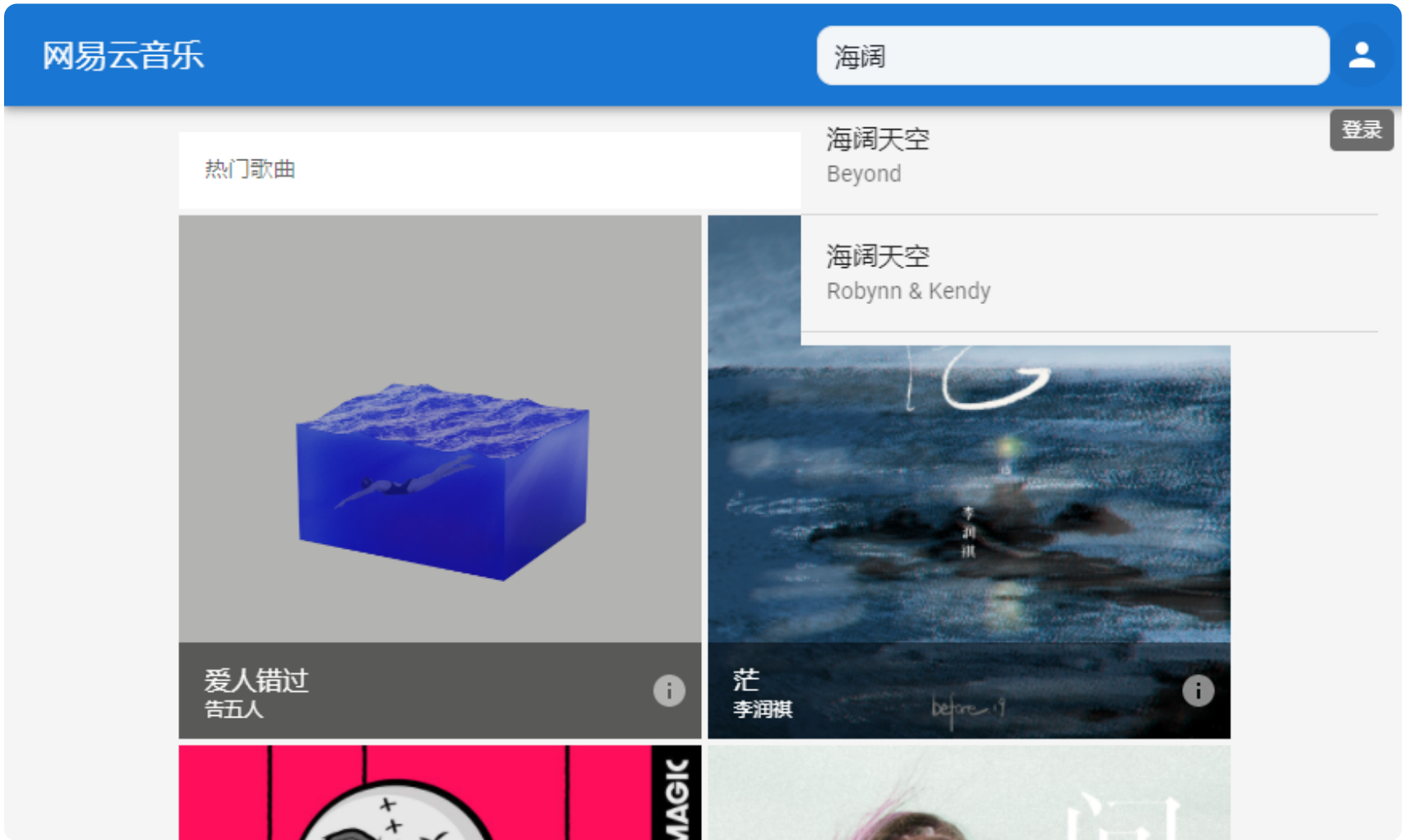


功能介绍

整体框架

网页由两部分组成：顶部导航栏和页面窗体

- 顶部导航栏: 搜索 + 用户状态
 - 搜索栏
 - 用户状态信息
- 页面窗体
 - 首页热榜
 - 歌曲详情
 - 搜索列表
 - 登录窗口



功能介绍

搜索框

```
1  useEffect(() => {
2      if (search === '') {
3          setSuggest([]);
4          return;
5      }
6      let newFetch = true;
7      const id = setTimeout(() => {
8          api
9              .getSearchSuggestion({ keywords: search })
10             .then((res) => {
11                 if (newFetch) {
12                     if (res.songs === undefined)
13                         setSuggest([]);
14                     else
15                         setSuggest(res.songs);
16                 }
17             })
18             .catch((e) => {
19                 console.log(e);
20                 if (newFetch) setSuggest([]);
21             });
22     }, 500);
23     return () => {
24         clearTimeout(id);
25         newFetch = false;
26     };
27 }, [search]);
```


功能介绍

搜索框

搜索框实现了发送请求时使用了如下技术

```
1  useEffect(() => {
2      if (search === '') {
3          setSuggest([]);
4          return;
5      }
6      let newFetch = true;
7      const id = setTimeout(() => {
8          api
9              .getSearchSuggestion({ keywords: search })
10             .then((res) => {
11                 if (newFetch) {
12                     if (res.songs === undefined)
13                         setSuggest([]);
14                     else
15                         setSuggest(res.songs);
16                 }
17             })
18             .catch((e) => {
19                 console.log(e);
20                 if (newFetch) setSuggest([]);
21             });
22     }, 500);
23     return () => {
24         clearTimeout(id);
25         newFetch = false;
26     };
27 }, [search]);
```

功能介绍

搜索框

搜索框实现了发送请求时使用了如下技术

1. 防抖技术
2. 先发后至技术

```
1  useEffect(() => {
2      if (search === '') {
3          setSuggest([]);
4          return;
5      }
6      let newFetch = true;
7      const id = setTimeout(() => {
8          api
9              .getSearchSuggestion({ keywords: search })
10             .then((res) => {
11                 if (newFetch) {
12                     if (res.songs === undefined)
13                         setSuggest([]);
14                     else
15                         setSuggest(res.songs);
16                 }
17             })
18             .catch((e) => {
19                 console.log(e);
20                 if (newFetch) setSuggest([]);
21             });
22         }, 500);
23     return () => {
24         clearTimeout(id);
25         newFetch = false;
26     };
27 }, [search]);
```

功能介绍

搜索框

搜索框实现了发送请求时使用了如下技术

1. 防抖技术

2. 先发后至技术

- 在用户不再输入过了500ms后才发送请求
- 若后一个请求比前一个请求先返回，前一个请求会失效

```
1  useEffect(() => {
2      if (search === '') {
3          setSuggest([]);
4          return;
5      }
6      let newFetch = true;
7      const id = setTimeout(() => {
8          api
9              .getSearchSuggestion({ keywords: search })
10             .then((res) => {
11                 if (newFetch) {
12                     if (res.songs === undefined)
13                         setSuggest([]);
14                     else
15                         setSuggest(res.songs);
16                 }
17             })
18             .catch((e) => {
19                 console.log(e);
20                 if (newFetch) setSuggest([]);
21             });
22         }, 500);
23         return () => {
24             clearTimeout(id);
25             newFetch = false;
26         };
27     }, [search]);
```

功能介绍

搜索框

搜索框实现了发送请求时使用了如下技术

1. 防抖技术

2. 先发后至技术

- 在用户不再输入过了500ms后才发送请求
- 若后一个请求比前一个请求先返回，前一个请求会失效

在搜索框中，若用户按 enter 键则跳转到搜索详情页

```
1  useEffect(() => {
2      if (search === '') {
3          suggest([]);
4          return;
5      }
6      let newFetch = true;
7      const id = setTimeout(() => {
8          api
9              .getSearchSuggestion({ keywords: search })
10             .then((res) => {
11                 if (newFetch) {
12                     if (res.songs === undefined)
13                         suggest([]);
14                     else
15                         suggest(res.songs);
16                 }
17             })
18             .catch((e) => {
19                 console.log(e);
20                 if (newFetch) suggest([]);
21             });
22         }, 500);
23         return () => {
24             clearTimeout(id);
25             newFetch = false;
26         };
27     }, [search]);
```

功能介绍

搜索框

搜索框实现了发送请求时使用了如下技术

1. 防抖技术

2. 先发后至技术

- 在用户不再输入过了500ms后才发送请求
- 若后一个请求比前一个请求先返回，前一个请求会失效

在搜索框中，若用户按 enter 键则跳转到搜索详情页

```
1  useEffect(() => {
2      if (search === '') {
3          setSuggest([]);
4          return;
5      }
6      let newFetch = true;
7      const id = setTimeout(() => {
8          api
9              .getSearchSuggestion({ keywords: search })
10             .then((res) => {
11                 if (newFetch) {
12                     if (res.songs === undefined)
13                         setSuggest([]);
14                     else
15                         setSuggest(res.songs);
16                 }
17             })
18             .catch((e) => {
19                 console.log(e);
20                 if (newFetch) setSuggest([]);
21             });
22         }, 500);
23         return () => {
24             clearTimeout(id);
25             newFetch = false;
26         };
27     }, [search]);
```

功能介绍

登录功能

用户凭据存放在localStorage中，自动根据token判断用户是否登录。

在鼠标经过右上角头像时支持显示登录用户的用户名。

! 账号/密码错误

手机号码 *
155xxxxxxxx

密码 *
.....

登录

注册账号

功能介绍

推荐歌曲

主页分别从热门歌单和“私人雷达”歌单中选取4首音乐，使用ImageList展示。

点击可以进入歌曲详情页。

功能介绍

搜索页

搜索详情页实现了分页功能，支持查看歌曲名、歌手名、专辑名和歌曲长度。

荷塘月色/徐一零 — 专辑名称：风止思难十一 — 歌曲时长：03:09

荷塘月色
睡不饱的老笨蛋 — 专辑名称：关于冬天的纪念 — 歌曲时长：00:36

没收
荷塘月色/童熹 — 专辑名称：没收 — 歌曲时长：03:08

荷塘月色
顾晚 — 专辑名称：G — 歌曲时长：02:19

如风过境
荷塘月色/徐一零/康博文/溇恩Seiwen — 专辑名称：如风过境 — 歌曲时长：02:53

荷塘月色
林文增 — 专辑名称：中国音乐大全.笛子卷 — 歌曲时长：05:44

功能介绍

歌曲详情页

歌曲详情页支持查看歌词和评论

评论也支持分页展示，展示了用户头像，用户名、评论和评论时间。

