
黑马程序员™
www.itheima.com
传智播客旗下
高端IT教育品牌
好客租房移动Web（上）

目录 Contents

1. 项目准备

1.1 项目介绍

- 好客租房 - 移动 Web 端
- 项目介绍：本项目是一个在线租房项目，实现了类似链家等项目的功能，解决了用户租房的需求
- 核心业务：在线找房（地图、条件搜索）、用户登录、房源发布等

1. 项目准备

1.1 项目介绍

技术栈

- React 核心库：react、react-dom、react-router-dom
- 脚手架：create-react-app
- 数据请求：axios
- UI组件库： antd-mobile
- 其他组件库： react-virtualized、formik+yup、react-spring 等
- 百度地图API

1. 项目准备

1.2 项目搭建

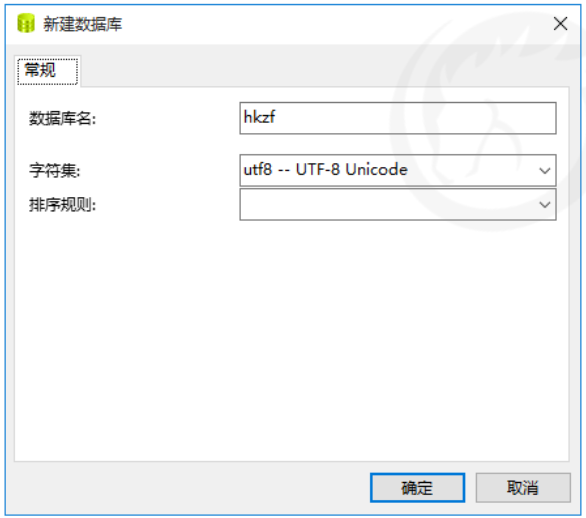
步骤

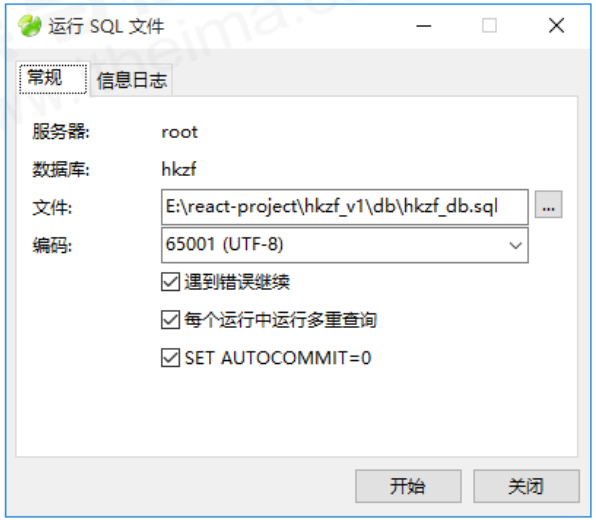
1. 本地接口部署（数据库、接口）
2. 使用脚手架初始化项目

1.2 项目搭建

本地接口部署

1. 创建并导入数据：数据库名称 hkzf（固定名称）
2. 启动接口：在 API 目录中执行 npm start
3. 测试接口：接口地址 http://localhost:8080



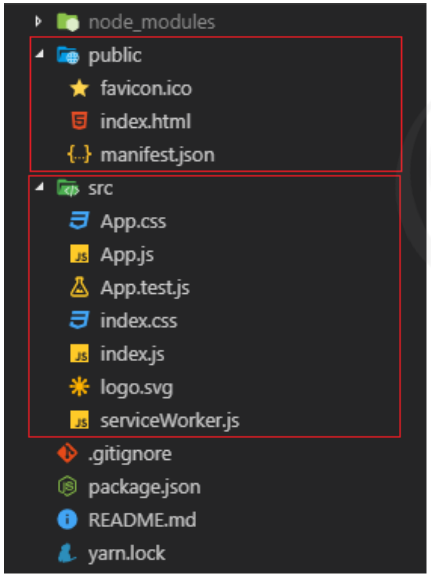


1. 项目准备

1.2 项目搭建

初始化项目

1. 初始化项目：**npx create-react-app hkzf-mobile**
2. 启动项目，在项目根目录执行命令：**yarn start**



public/	公共资源
index.html	首页（必须有）
manifest.json	PWA应用的元数据
src/	项目源码，写项目功能代码
index.js	项目入口文件（必须有）
App.js	项目的根组件
App.test.js	App组件的测试文件
serviceWorker.js	用来实现PWA（可选）

1.2 项目搭建

初始化项目

1. 初始化项目：**npx create-react-app hkzf-mobile**
2. 启动项目，在项目根目录执行命令：**yarn start**
3. 调整项目中 src 目录结构如下：

```
src/                    项目源码，写项目功能代码
  assets/               资源（图片、字体图标等）
  components/           公共组件
  pages/                页面
  utils/                工具
  App.js                根组件（配置路由信息）
  index.css             全局样式
  index.js              项目入口文件（渲染根组件、导入组件库等）
```

1.3 组件库 antd-mobile

介绍和使用

1. 打开 antd-mobile 的文档
2. 安装：yarn add antd-mobile
3. 在 App.js 根组件中导入要使用的组件
4. 渲染组件
5. 在 index.js 中导入组件库样式

```
import { Button } from 'antd-mobile'
```

```
<Button />
```

```
import 'antd-mobile/dist/antd-mobile.css'
```

1.4 配置基础路由

步骤

1. 安装：yarn add react-router-dom
2. 导入路由组件：Router / Route / Link
3. 在 pages 文件夹中创建 Home/index.js 和 CityList/index.js 两个组件
4. 使用 Route 组件配置首页和城市选择页面

```
import { BrowserRouter as Router, Route, Link } from 'react-router-dom'
```

```
<Route path="/home" component={Home} />
<Route path="/citylist" component={CityList} />
```

1. 项目准备

1.5 外观和样式调整

步骤

1. 修改页面标题：在 index.html 中修改
2. 基础样式调整：在 index.css 中修改

```
html,
body {
  height: 100%;
  font-family: 'Microsoft YaHei';
  color: #333;
  background-color: #fff;
}
* {
  box-sizing: border-box;
}
```

目录 Contents

2. 项目整体布局

2.1 两种布局页面

分析

1. 有 TabBar 的页面：首页、找房、资讯、我的。
2. 无 TabBar 的页面：城市选择等（简单，不需要额外处理）。

2. 项目整体布局

2.1 两种布局页面

分析

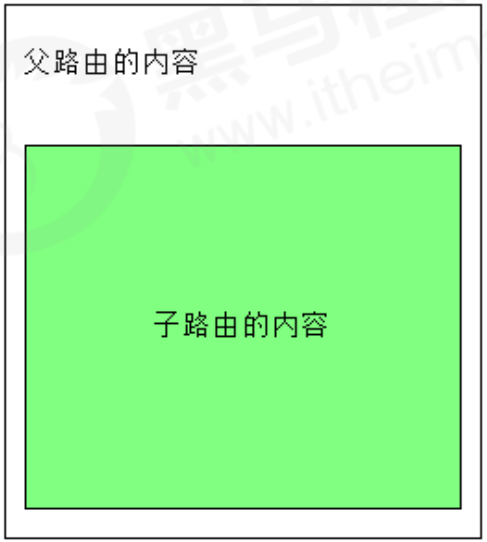
1. 有 TabBar 的页面：首页、找房、资讯、我的。
2. 无 TabBar 的页面：城市列表等（简单，不需要额外处理）。
3. TabBar 的菜单也可以实现路由切换，也就是在路由内部切换路由（嵌套路由）

2. 项目整体布局

2.2 嵌套路由

说明

- 嵌套路由：路由内部包含路由
- 用 Home 组件表示父路由的内容
- 用 News 组件表示子路由的内容



2. 项目整体布局

2.2 嵌套路由

使用步骤

1. 在 pages 文件夹中创建 News/index.js 组件
2. 在 Home 组件中，添加一个 Route 作为子路由（嵌套的路由）的出口
3. 设置嵌套路由的 path，格式以父路由 path 开头（父组件展示，子组件才会展示）
4. 修改 pathname 为 /home/news，News 组件的内容就会展示在 Home 组件中了

```
<Router>
  <div>
    <Route path="/home" component={Home} />
  </div>
</Router>
```

```
const Home = () => (
  <div>
    <Route path="/home/news" component={News} />
  </div>
)
```

2.3 实现 TabBar

1. 基本使用

① 打开 antd-mobile 组件库中 TabBar 组件的 文档。

② 选择 APP 型选项卡菜单，点击（</>）显示源码。

③ 拷贝核心代码到 Home 组件中。

④ 分析并调整代码，让其能够在项目中正常运行。

2.3 实现 TabBar

2. 修改 TabBar 外观样式

① 删除前面路由的演示代码。

② 修改 TabBar 菜单项文字标题。

③ 修改 TabBar 菜单文字标题颜色（选中和未选中）。

④ 使用字体图标，修改 TabBar 菜单的图标。

⑤ 修改 TabBar 菜单项的图标大小。

⑥ 调整 TabBar 菜单的位置，让其固定在最底部。

2.3 实现 TabBar

3. TabBar 配合路由使用

① 根据 TabBar 组件 文档 设置不渲染内容部分（只保留菜单项，不显示内容）。

```
<TabBar noRenderContent={true}></TabBar>
```

2.3 实现 TabBar

3. TabBar 配合路由使用

① 根据 TabBar 组件 文档 设置不渲染内容部分（只保留菜单项，不显示内容）。

② 给 TabBar.Item 绑定点击事件。

③ 在点击事件中调用 history.push() 实现路由切换。

④ 创建 TabBar 组件菜单项对应的其他 3 个组件，并在 Home 组件中配置路由信息。

```
<TabBar.Item
  onPress={() => {
    this.props.history.push('/home/list')
  }}
/>
```

2.3 实现 TabBar

3. TabBar 配合路由使用

① 根据 TabBar 组件 文档 设置不渲染内容部分（只保留菜单项，不显示内容）。

② 给 TabBar.Item 绑定点击事件。

③ 在点击事件中调用 history.push() 实现路由切换。

④ 创建 TabBar 组件菜单项对应的其他 3 个组件，并在 Home 组件中配置路由信息。

⑤ 给菜单项添加 selected 属性，设置当前匹配的菜单项高亮。

```
selectedTab = this.props.location.pathname

<TabBar.Item
  selected={selectedTab === '/home/list'}
  onPress={() => {
    this.setState({
      selectedTab: '/home/list'
    })
 }}
/>
```

2.3 实现 TabBar

4. TabBar.Item 代码重构

① 提供菜单数据（包含菜单项的特有信息）。

② 使用 map 方法，遍历数据，渲染 TabBar.Item。

```
const tabItems = [
  {
    title: '首页',
    icon: 'icon-ind',
    path: '/home/index'
  }
]
```

```
tabItems.map(item => <TabBar.Item ... />)
```

目录 Contents

3. 首页模块

3.1 首页路由处理

说明

- 修改首页路由规则为： /home（去掉 /index）
- 配合默认路由，实现默认跳转到 /home（路由文档）
- render 属性：是一个函数prop，用于指定要渲染的内容
- Redirect 组件用于实现路由重定向，to 属性指定要跳转到的路由地址

```
<Route exact path="/home" component={Index} />
```

```
<Route exact path="/" render={() => <Redirect to="/home" />} />
```

3. 首页模块

3.2 轮播图

1. 基本使用

① 打开 antd-mobile 组件库的 Carousel 组件文档。

② 选择基本，点击（</>）显示源码。

③ 拷贝核心代码到 Index 组件中。

④ 分析并调整代码，让其能够在项目中正常运行。

3.2 轮播图

2. 获取轮播图数据

① 安装 axios：yarn add axios 。

② 在 Index 组件中导入 axios 。

③ 在 state 中添加轮播图数据：swipers 。

④ 新建一个方法 getSwipers 用来获取轮播图数据，并更新 swipers 状态。

⑤ 在 componentDidMount 钩子函数中调用该方法。

⑥ 使用获取到的数据渲染轮播图。

3. 首页模块

3.3 导航菜单

步骤

1. 打开 Flex 布局组件文档
2. 使用 Flex 布局，创建导航菜单结构

3.4 在脚手架中使用Sass

步骤

① 打开脚手架的文档，找到添加Sass样式

② 安装Sass： yarn add node-sass

③ 创建后缀名称为 .scss 或 .sass 的样式文件

④ 在组件中导入Sass样式文件

3.5 租房小组

1. 业务介绍

- 需求：根据当前地理位置展示不同小组信息
- 需要后台接口根据用户找房数据，推荐用户最感兴趣的内容，核心业务在后端
- 前端只负责展示数据

3.5 租房小组

2. 数据获取

① 在 state 中添加租房小组数据：groups 。

② 新建一个方法 getGroups 用来获取数据，并更新 groups 状态。

③ 在 componentDidMount 钩子函数中调用该方法。

④ 使用获取到的数据渲染租房小组数据。

3.5 租房小组

3. 结构和样式

① 实现标题的结构和样式。

② 打开 Grid 宫格组件文档。

③ 选择 基本 菜单，点击（</>）显示源码。

④ 拷贝核心代码到 Index 组件中。

⑤ 分析并调整代码，让其能够在项目中正常运行。

3.6 顶部导航

步骤

1. 实现结构和样式。
2. 添加城市选择、搜索、地图找房页面的路由跳转。

3.7 H5 中的地理位置 API

- 场景：根据当前地理位置，获取当前所在城市的房源信息
- 作用：在 Web 应用程序中获取地理位置（文档）
- 说明：地理位置 API 通过 navigator.geolocation 对象提供，通过 getCurrentPosition 方法获取
- 注意：获取到的地理位置跟 GPS、IP地址、WIFI和蓝牙的MAC地址、GSM/CDMS的ID 有关
- 比如：手机优先使用GPS定位，笔记本等最准确的定位是WIFI

```
navigator.geolocation.getCurrentPosition(position => {
  // position对象表示当前位置信息
  // 常用： latitude 纬度 / longitude 经度
  // 知道： accuracy 经纬度的精度 / altitude 海拔高度 / altitudeAccuracy 海拔高度的精度 / heading 设备行进方向 / speed 速度
})
```

3.8 百度地图 API

1. 介绍

- H5 的地理位置 API 只能获取到经纬度信息
- 实际开发中，会使用百度地图/高德地图来完成地理位置的相关功能
- 租房项目中，通过百度地图 API 实现地理定位和地图找房功能
- 百度地图文档 （首页 → 开发者文档 → JavaScript API）
- 注意：使用前，需要先申请 百度账号和 ak ，获取到的 ak

3.8 百度地图 API

2. 申请百度账号和密钥

① 注册百度账号，登录百度地图开放平台。

② 点击创建应用。

③ 获取到密钥（ak）。

3.8 百度地图 API

3. 使用步骤

① 引入百度地图 API 的JS文件，替换自己申请好的密钥。

```
<script src="http://api.map.baidu.com/api?v=3.0&ak=您的密钥"></script>
```

② 在 index.css 中设置全局样式

```
html, body, #root, .App {
  height:100%
}
body{
  margin:0px;
  padding:0px
}
```

3.8 百度地图 API

3. 使用步骤

③ 创建 Map 组件，配置路由。并在 Map 组件中，创建地图容器元素，并设置样式

```
// Map组件
<div className="map">
  <div id="container"></div>
</div>

// 设置样式：
.map, #container {
  height:100%
}
```

3.8 百度地图 API

3. 使用步骤

④ 创建地图实例

```
const map = new BMap.Map("container")
```

⑤ 设置中心点坐标。

```
const point = new BMap.Point(116.404, 39.915)
```

⑥ 初始化地图，同时设置展示级别。

```
map.centerAndZoom(point, 15)
```

3.8 百度地图 API

4. 获取顶部导航城市信息

① 打开百度地图JS API 定位文档 。

② 通过 IP 定位获取到当前城市名称。

③ 调用我们服务器的接口，换取项目中的城市信息（有房源的城市的名称和id）。

④ 将接口返回的城市信息展示在顶部导航栏中。

```
const myCity = new BMap.LocalCity()
myCity.get((res) => {
  const cityName = res.name
  console.log('当前定位城市名称为：', cityName)
})
```

```
axios.get(`/area/info?name=${cityName}`)
```

目录 Contents

4. 城市选择模块

4.1 功能分析

- 业务：切换城市，查看该城市下的房源信息
- 功能：1. 顶部导航栏 2. 城市列表展示 3. 使用索引快速切换城市 4. 点击城市名称切换城市
- 第三方组件： react-virtualized 长列表
- 难点：数据格式处理、react-virtualized组件在项目中的使用

4. 城市选择模块

4.2 顶部导航栏

步骤

1. 打开 antd-mobile 组件库的 NavBar 导航栏组件 文档。
2. 从文档中拷贝组件示例代码到项目中，让其正确运行。
3. 修改导航栏样式和结构。

4. 城市选择模块

4.3 获取并处理城市列表数据

步骤

1. 页面加载时，根据接口获取城市列表数据。
2. 分析当前数据格式以及该功能需要的数据格式。
3. 转化当前数据格式为所需数据格式。
4. 获取热门城市数据，并添加到现有数据列表中（顺序）。
5. 获取当前定位城市数据，并添加到现有数据列表中。

```
// 接口返回的数据格式：
[{ "label": "北京", "value": "", "pinyin": "beijing", "short": "bj" }]
// 渲染城市列表的数据格式为：
{ a: [{}, {}], b: [{}, ...] }
// 渲染右侧索引的数据格式为：
['a', 'b']
```

4. 城市选择模块

4.3 获取并处理城市列表数据

当前定位城市数据

- 问题：首页模块中，需要获取定位城市，城市列表模块中也需要获取定位城市，该如何处理？
- 答案：封装成函数，哪个页面要获取定位城市，直接调用该方法即可。

4. 城市选择模块

4.3 获取并处理城市列表数据

当前定位城市数据

1. 在 utils 目录中，新建 index.js，在该文件中封装
2. 创建并导出获取定位城市的函数 getCurrentCity
3. 判断 localStorage 中是否有定位城市
4. 如果没有，就使用首页中获取定位城市的代码来获取，并且存储到本地存储中，然后返回该城市数据
5. 如果有，直接返回本地存储中的城市数据

4. 城市选择模块

4.4 长列表性能优化

1. 概述

- 场景：展示大型列表和表格数据（比如：城市列表、通讯录、微博等），会导致页面卡顿、滚动不流畅等性能问题
- 产生性能问题的原因：大量 DOM 节点的重绘和重排
- 其他原因：老旧设备
- 其他问题：移动设备耗电加快、影响移动设备电池寿命
- 优化方案：1. 懒渲染 2. 可视区域渲染

4.4 长列表性能优化

2. 懒渲染说明

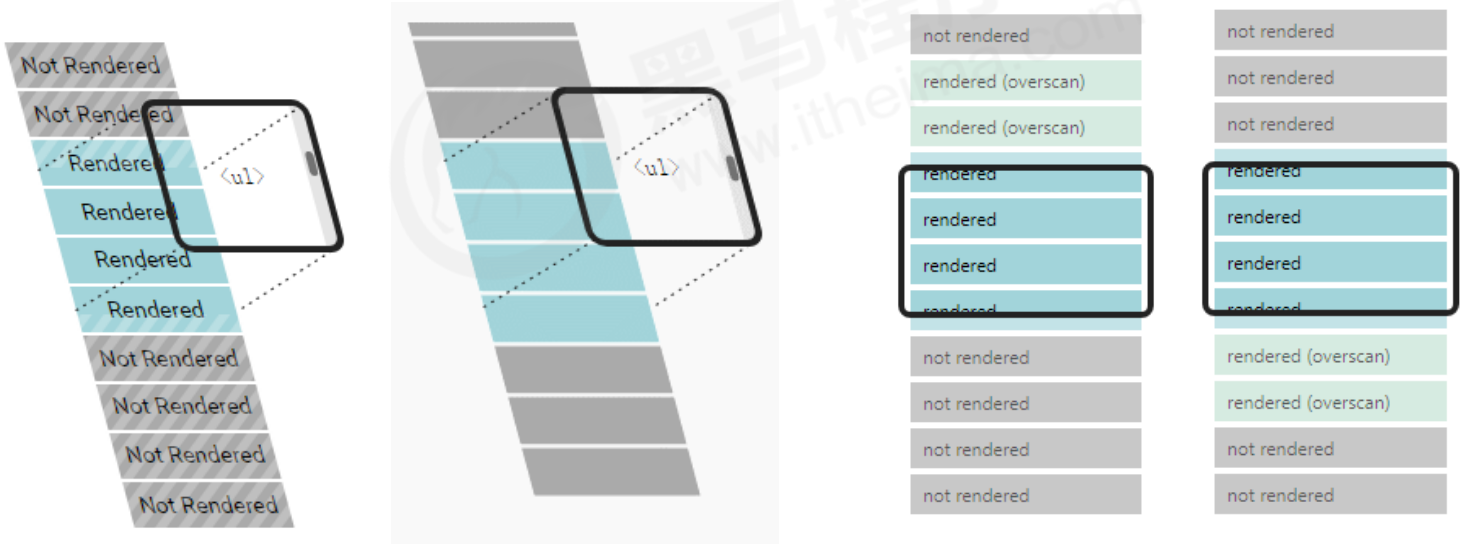
- 常见的长列表优化方案，常见于移动端
- 原理：每次只渲染一部分（比如10条数据），等渲染的数据即将滚动完时，再渲染下面部分
- 优点：每次渲染一部分数据，速度快
- 缺点：数据量大时，页面中依然存在大量 DOM 节点，占用内存过多、降低浏览器渲染性能，导致页面卡顿
- 使用场景：数据量不大的情况（比如 1000 条，具体还要看每条数据的复杂程度）

4. 城市选择模块

4.4 长列表性能优化

3. 可视区域渲染（react-virtualized）

- 原理：只渲染页面可视区域的列表项，非可视区域的数据“完全不渲染”，在滚动列表时动态更新列表项
- 使用场景：一次性展示大量数据的情况（比如：大表格、微博、聊天应用等）



4.5 react-virtualized

1. 概述

① 在项目中的应用：实现 城市选择 列表页面的渲染。

② react-virtualized 是 React 组件，用来高效渲染大型列表和表格数据。

③ GitHub地址：react-virtualized 。

④ 功能类似的轻量级组件：react-window 。

4. 城市选择模块

4.5 react-virtualized

2. 基本使用

① 安装：yarn add react-virtualized 。

② 在项目入口文件 index.js 中导入样式文件（只导入一次即可）。

③ 打开文档，点击 List 组件，进入 List 的文档中。

④ 翻到文档最底部，将示例代码拷贝到项目中。

⑤ 分析示例代码。

```
import 'react-virtualized/styles.css'
```

4. 城市选择模块

4.6 渲染城市列表

1. 让List组件占满屏幕

① 打开 AutoSizer 高阶组件的文档。

② 导入 AutoSizer 组件。

③ 通过 render-props 模式，获取到 AutoSizer 组件暴露的 width 和 height 属性。

④ 设置 List 组件的 width 和 height 属性。

⑤ 设置城市选择页面根元素高度 100%，让 List 组件占满整个页面。

⑥ 调整样式，让页面不要出现全局滚动条，避免顶部导航栏滚动。

```
<AutoSizer>
  {({ height, width }) => (
    <List height={height} width={width} />
  )}
</AutoSizer>
```

4. 城市选择模块

4.6 渲染城市列表

2. 使用List组件渲染列表

① 将获取到的 cityList 和 cityIndex 添加为组件的状态数据。

② 修改 List 组件的 rowCount 为 cityIndex 数组的长度。

③ 将 rowRenderer 函数，添加到组件中，以便在函数中获取到状态数据 cityList 和 cityIndex。

④ 修改 List 组件的 rowRenderer 为组件中的 rowRenderer 方法。

⑤ 修改 rowRenderer 方法中渲染的每行结构和样式。

⑥ 修改 List 组件的 rowHeight 为函数，动态计算每一行的高度（因为每一行高度都不相同）。

4.7 城市索引列表

1. 渲染城市索引列表

① 封装 renderCityIndex 方法，用来渲染城市索引列表。

② 在方法中，获取到索引数组 cityIndex ， 遍历 cityIndex ， 渲染索引列表。

③ 将索引 hot 替换为 热。

④ 在 state 中添加状态 activeIndex，来指定当前高亮的索引。

⑤ 在遍历 cityIndex 时，添加当前字母索引是否高亮的判断条件。

4.7 城市索引列表

2. 滚动城市列表让对应索引高亮

① 给 List 组件添加 onRowsRendered 配置项，用于获取当前列表渲染的行信息。

② 通过参数 startIndex 获取到，起始行索引（也就是城市列表可视区最顶部一行的索引号）。

③ 判断 startIndex 和 activeIndex 是否相同（判断的目的是为了提升性能，避免不必要的 state 更新）。

④ 当 startIndex 和 activeIndex 不同时，更新状态 activeIndex 为 startIndex 的值。

4. 城市选择模块

4.7 城市索引列表

3. 点击索引置顶该索引城市

① 给索引列表项绑定点击事件。

② 在点击事件中， 通过 index 获取到当前项索引号。

③ 调用 List 组件的 scrollToRow 方法，让 List 组件滚动到指定行。

④ 设置 List 组件的 scrollToAlignment 配置项值为 start，保证被点击行出现在页面顶部。

4. 城市选择模块

4.8 切换城市

① 给城市列表项绑定点击事件。
② 判断当前城市是否有房源数据（只有北/上/广/深四个城市有数据）。
③ 如果有房源数据，则保存当前城市数据到本地缓存中，并返回上一页。
④ 如果没有房源数据，则提示用户：该城市暂无房源数据，不执行任何操作。

好客租房移动Web（上）

1. 项目准备：部署本地接口、脚手架初始化项目、antd-mobile、路由等。
2. 项目整体布局：分析两种页面布局，使用嵌套路由实现带 TabBar 页面布局等。
3. 首页模块：租房小组结构布局、数据获取、H5 地理定位和百度地图地理定位等。
4. 城市选择模块：数据结构处理、长列表性能优化、react-virtualized、索引列表等。

传智播客旗下高端IT教育品牌