

一：前期学习

1.1 基础：html CSS JavaScript

一个基本网站包含许多网页，而一个网页由 html, css 和 javascript 组成。其中 html 是主体，装载各种 DOM 元素（每一对 HTML 标签都是一个 DOM 元素）；css 用来装饰 DOM 元素；javascript 控制 DOM 元素。

html

html，超文本标记语言，标记内容，通过各种（DOM）元素为内容添加语义结构（层次，关系和含义）。浏览器解析 HTML 并生成 DOM 后，对 DOM 应用视觉规则并将像素绘制到屏幕，也即渲染。

CSS

css，层叠样式表，控制 DOM 元素的视觉外观。可在 html 的头元素<head>里书写，或者书写在 CSS 文件并在 html 里引用该文件。css 由选择器，一条或多条声明两个主要部分组成。

Javascript

Javascript 是动态脚本语言，通过操作 DOM 动态修改页面。它是连接前台（html）和后台服务器的桥梁。JavaScript 能够改变 HTML 属性、HTML 样式 (CSS)和 HTML 元素，亦可实现对事件的反应。

1.2 进阶：React Mobx AntDesign

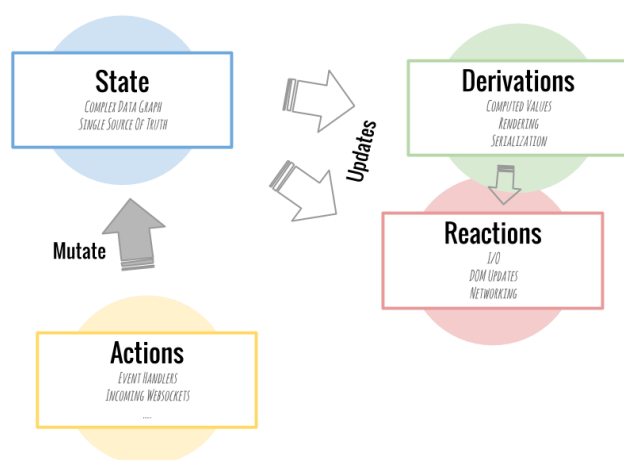
React

React 是一个用于构建用户界面的 JavaScript 库。它使用组件化开发方式，我们可以在 React 中配合使用 JSX（一个 JavaScript 的语法扩展）。JSX 可以生成 React “元素”。同时，使用淘宝定制的 cnpm 命令行工具代替默认的 npm 构建 react 的开发环境。

mobx

在 react 中还可以使用 mobx，它是一个使用函数式编程的状态管理库（比 redux 更加友好的版本），在整个数据流中，通过事件驱动（UI 事件、网络请求...）触发 Actions，在 Actions 中修改了 State 中的值，这里的 State 也即应用中的 store 树（存储数据），然后根据新的 State 中的数据计算出所需要的计算属性值，最后响应到 UI 视图层。

mobx 数据流图解：



mobx 也可以使用 cnpm 安装然后引入。

AntDesign

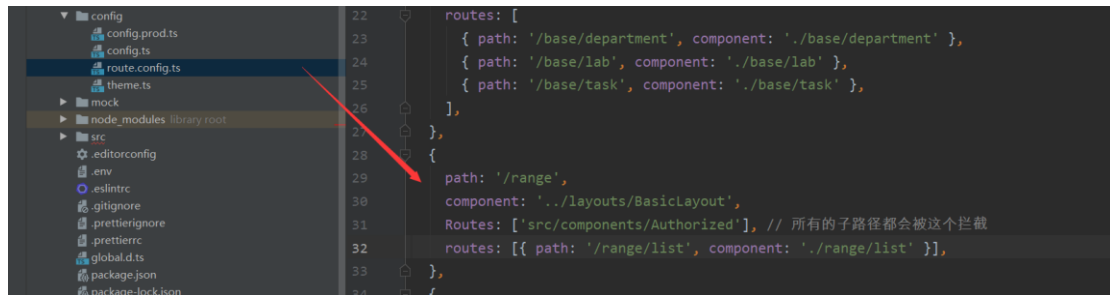
一个功能强大的高质量 React 组件库，使用 TypeScript 开发。此项目开发的许多组件从官方网站生成代码并改动使用，包括按钮、布局、面包屑、导航栏、下拉菜单等。

二：开发流程：

step1：前端路由配置

通过配置 js 文件，根据用户请求的 url 地址导航到具体的内容或页面。在访问一个新页面的时候仅仅变换路径，使用浏览器的前进，后退键的时候会重新发送请求。

实现示例：



path

Type: string

配置路径通配符。

component

Type: string

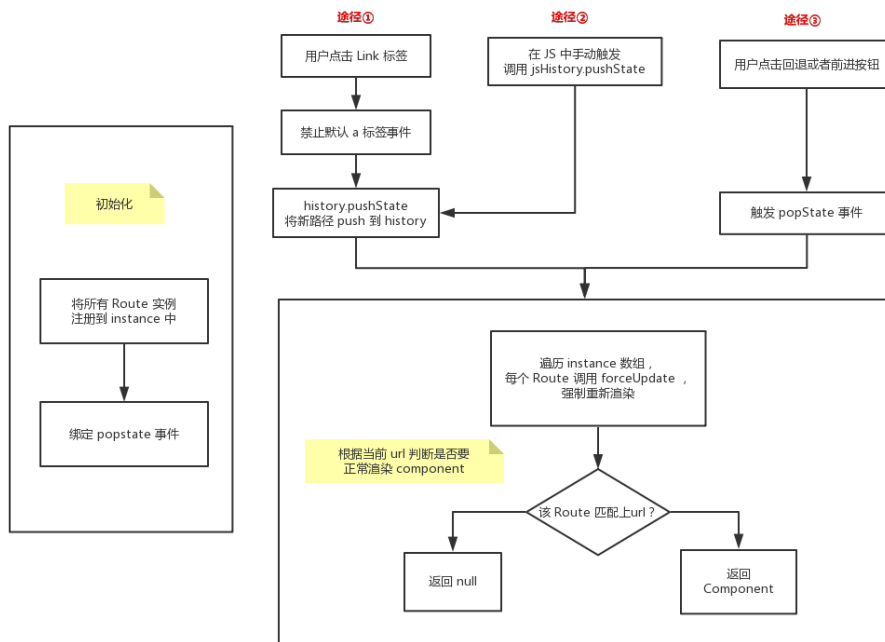
配置 location 和 path 匹配后用于渲染的 React 组件路径。绝对/相对路径，如果是相对路径，会从 src/pages 开始找起。

如果指向 src 目录的文件，可以用 @，也可以用 ../。比如 component: '@/layouts/basic', 或者 component: '../layouts/basic', 一般用前者。

routes

配置子路由，然后在 src/...中渲染子路由

前端路由的基本流程图（参考链接：<https://www.jianshu.com/p/d2aa8fb951e4>）



step2: 配置网页布局-layout

layout 之 Breadcrumb 面包屑配置

breadcrumb 用于显示当前页面在系统层级结构中的位置，并能向上返回。

效果如图:

首页 / 比赛场地管理 / 场地列表

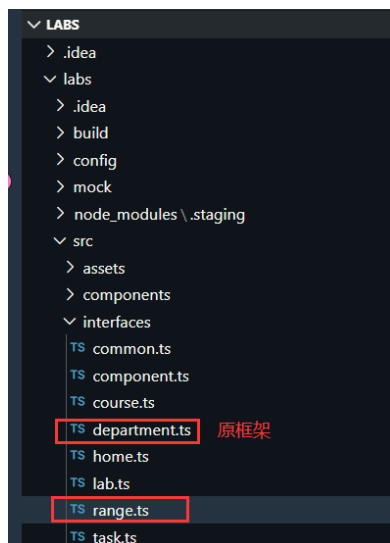


layout 之 导航栏内容配置

定义一个数组来描述页面左侧的导航栏。导航栏包括比赛场地管理、场地列表以及展开后的各项管理，不同的名称对应不同的 apiUrl。例如：

```
{
  resourceKey: 'range',
  apiUrl: '',
  icon: 'icon-homepage',
  description: '比赛场地管理',
  children: [
    {
      resourceKey: 'list',
      apiUrl: '/range/list',
      description: '场地列表',
    },
  ],
},
```

step3: 接口

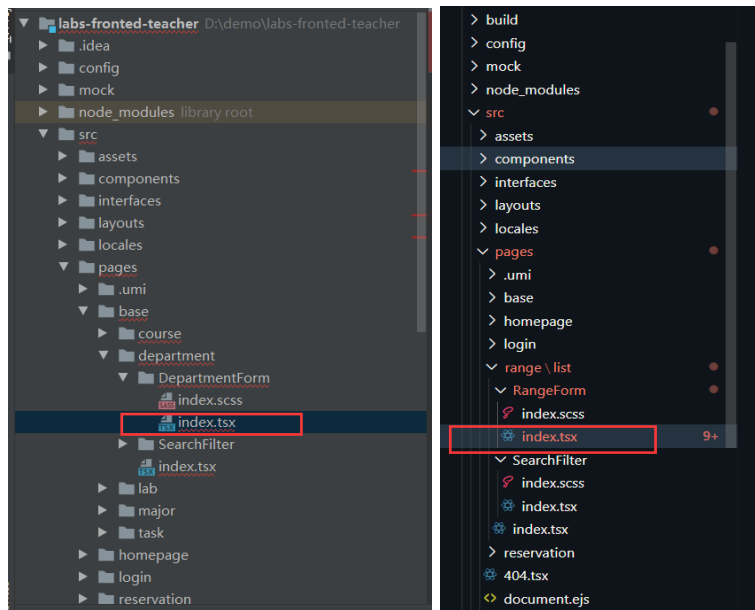


此处 export 一系列文档中给出的接口，在后续文件中按需调用。

step4: 项目代码

队伍信息输入的表单页面（新建/查询界面）

1. 文件位置：左为原始框架 右为本组框架



2. import:

```
import * as React from 'react';
import Form, { FormComponentProps } from 'antd/lib/form';
import { FILTER_FORM_LAYOUT } from '@constant';
import { Input, Select } from 'antd';
import styles from './index.scss';

import { compose, withState } from 'recompose';
import { RangeEditModel } from '@interfaces/range';
import ModalButtons from '@components/ModalButtons';
```

3. 内容:

使用 typescript 语法和 react 库，创建队伍信息输入表单。

由基本的表单数据域控制展示组成，包含布局、初始化、提交等操作。相关组件代码从 antd 生成使用，其中场地位置、名称采用文字输入框，场地状态采用多选下拉菜单。信息将输入并提交到后端数据库。

4. 效果:

新增场地

* 场地名称:

请输入

* 场地位置:

请输入

* 场地状态:

请选择场地状态

关闭原因:

请输入

备注:

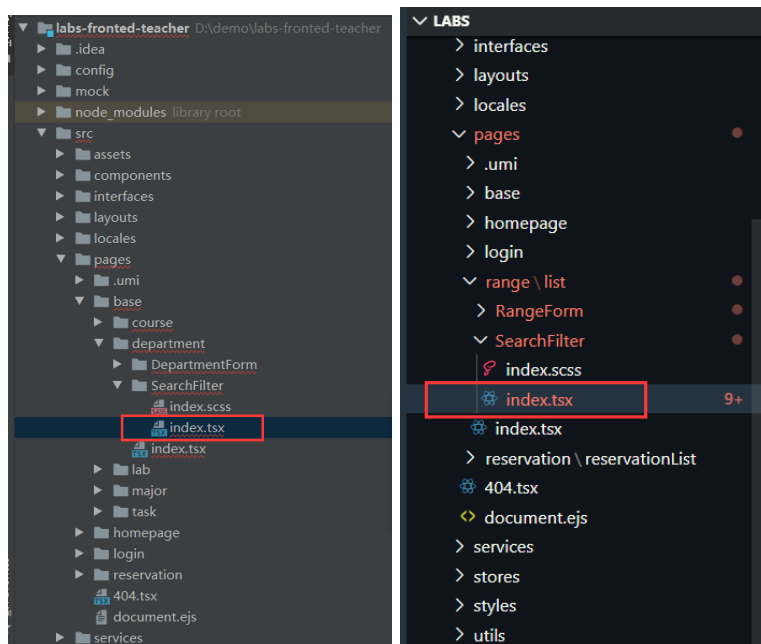
请输入备注

取消

保存

队伍信息查询页面的创建（首界面 1）：

1. 文件位置：



2. import:

```
import * as React from 'react';
import { Input, Button, Select } from 'antd';
import styles from './index.scss';
import { RangeSearchProps } from '@/interfaces/range.ts';
const { Option } = Select;
```

3. 内容：

同队伍信息输入的表单页面（新建/查询界面），场地名称、位置采用手动输入，场地状态采用下拉菜单选择，请求将提交到后端数据库实现 select 查询操作。

4. 效果：

场地名称：

请输入场地名称

场地位置：

请输入场地位置

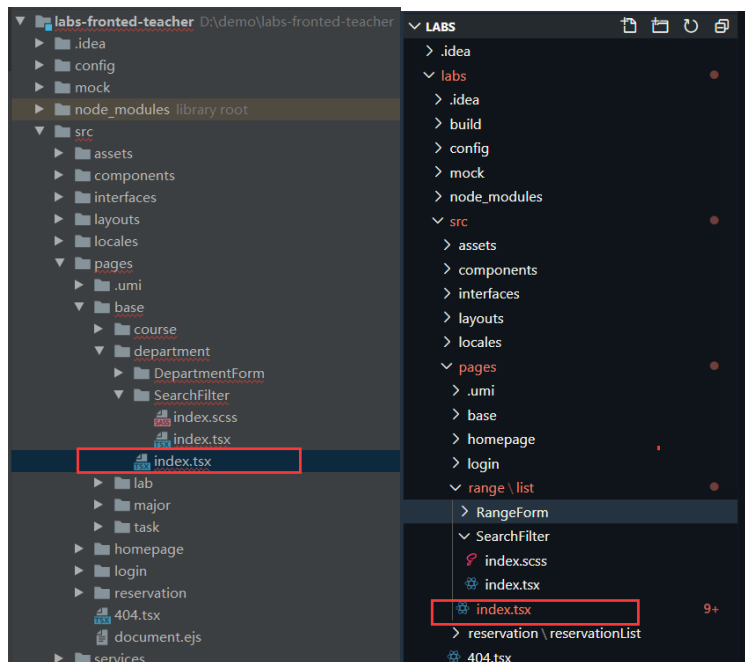
场地状态：

选择场地状态

查询

已有信息展示&增删改查（首界面 2）：

1. 文件位置：



2. import:


```
import * as React from 'react';
import { observer, inject } from 'mobx-react';
import {
  RangeModel,
  RangeEditModel,
  RangeSearchProps,
  defaultRangeSearchProps,
} from '@/interfaces/range';
import CustomTable from '@/components/CustomTable';
import { compose, withState } from 'recompose';
import { RangeService } from '@/services/range.service';
// import SearchFilter from './SearchFilter';
import Loading from '@/components/Loading';
import { Divider, Modal } from 'antd';
const CommonModal = React.lazy(() => import('@/components/CommonModal'));
import RangeForm from './RangeForm';
import { ButtonItem } from '@/interfaces/component';
import SearchFilter from './SearchFilter';
```

3. 内容：

在查询框下展示已有（输入）的队伍信息，提供修改和删除操作，并实现删除确认弹窗。将 store 取到的数据体现在最前端，

4. 效果：

场地名称:

请输入场地名称

场地位置:

请输入场地位置

场地状态:

选择场地状态

查询

选择	场地编码	场地名称	位置	状态编码	状态描述	更新时间	操作
<input type="checkbox"/>	RG2004110002	游泳馆(比赛进行中无法删除)	n12	2	进行中	2020-04-11 15:53:13	修改 删除
<input type="checkbox"/>	RG2004280005	射击馆	123	2	进行中	2020-04-28 19:10:12	修改 删除
<input type="checkbox"/>	RG2004280006	摔跤馆	string	1	未开始	2020-04-28 19:10:47	修改 删除
<input type="checkbox"/>	RG2004280007	体育馆	string	3	已结束	2020-04-28 19:11:06	修改 删除
<input type="checkbox"/>	RG2004280008	田径场	string	3	已结束	2020-04-28 19:11:18	修改 删除
<input type="checkbox"/>	RG2004280009	羽毛球馆	string	1	未开始	2020-04-28 19:11:42	修改 删除

新增

删除

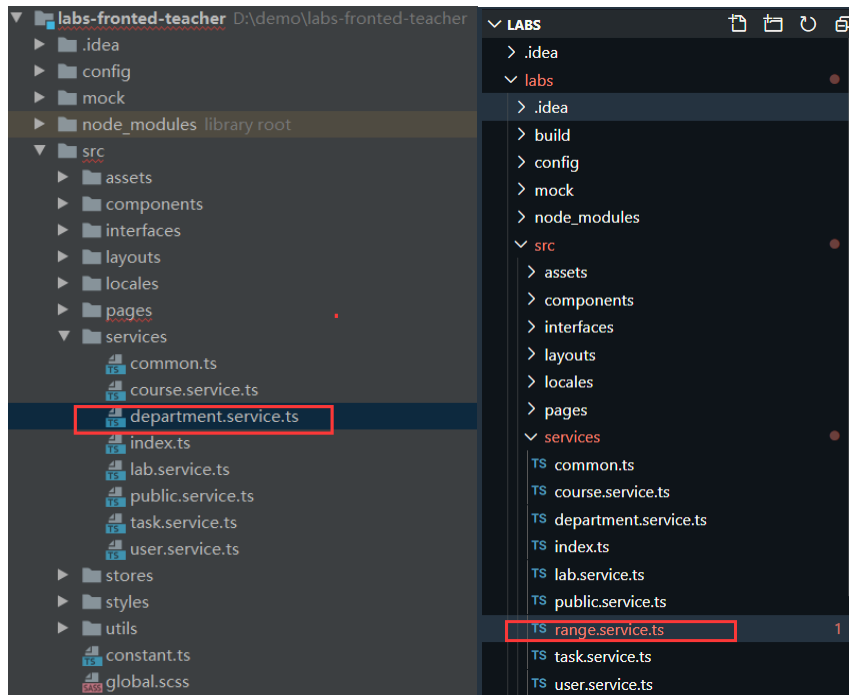
<

1

>

service 服务层的配置

1. 文件位置：



2. import:

```
import { action } from 'mobx';
import HttpClient from '../utils/HttpClient';
import { BACKEND_URL, messageFail } from './common';
import { Pagination, initialPaginationValue } from '@/interfaces/common';
import { RangeModel, RangeSearchProps, RangeEditModel } from '@/interfaces/range';
import { RangeStore } from '@/stores/range.store';
```

3. 内容:

Service 层主要负责业务模块的逻辑应用设计，设计接口和其实现的类，接着在 Spring 的配置文件中配置其实现的关联，即可在应用中调用 Service 接口来进行业务处理。

本文件包括页面数据、更新请求、数据添加、数据删除方法，判断成功或失败并通过对应的 URL 返回后端数据。

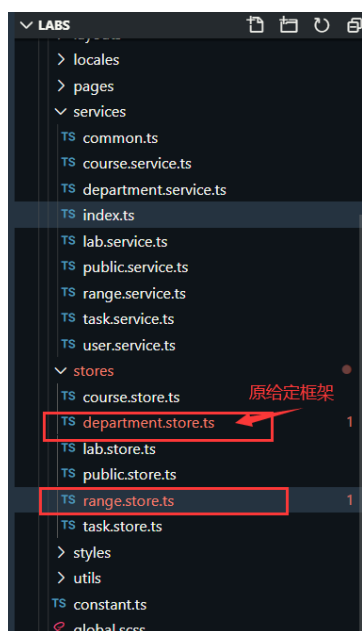
调用 mobx 的 action API，通过 @action 装饰函数应用于更改状态的这几个方法。

mobx 可以让现有的 js 数据结构(数组，对象)添加可观察功能。这个功能主要是指，能对数据的各种操作进行监听(例如数组的元素的变化，对象的属性的值的更改。)，从而实现当下前端所流行的响应式编程。流程图如下：



mobx-store 数据仓库

1. 文件位置:



2. 内容:

Store 主要将逻辑和状态从组件中移至一个独立单元，提供方法更新数据状态（部门分页列表数据），同时@observable 声明了 loading 和 pagedata 两个可观察对象，给 react 组件引入了可观察属性。在\src\services\range.service.ts 中再被 import。

另外需要明晰：store 是指在应用中存储数据的地方，而上面部分用到的 action 则是所有触发 store 数据变化的地方。