### React 脚手架搭建全过程

###### 安装 create-react-app

npx create-react-app react-mycli

cd react-mycli

yarn start (安装yarn 命令 : npm install -g yarn )

###### 安装 Ant Desigin

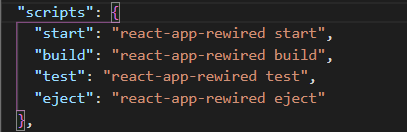
yarn add antd

###### 配置 Ant Design 按需加载

第一步：安装 react-app-rewired 为项目添加 webpack 配置，安装完成后 在项目的根目录中新建 config-overrides.js 文件；

yarn add react-app-rewired

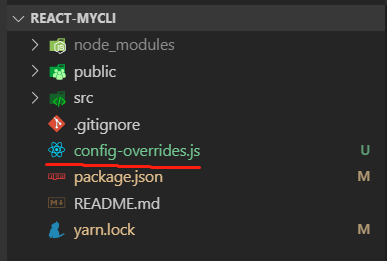
第二步：修改 package.json 文件，修改如下

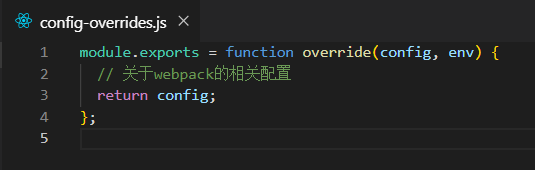


第三步：由于react-app-rewired 2.X 以后不在支持 injectBabelPlugin 的方式，需要安装 customize-cra

yarn add customize-cra

第四部：在根目录里添加 config-override.js 文件，如下所示：



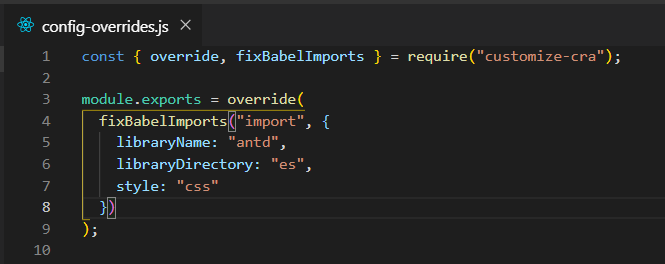


好了，基本的操作就是这样，下面我们再具体到项目中做一些配置

第五步：配置按需加载，我们需要使用 babel-plugin-import，它是一个用于按需加载组件代码和样式的babel插件

yarn add babel-plugin-import

接着配置webpack ，修改 config-overrides.js ，修改后如下所示



这样我们在组件中就可以按需引入组件库中的所需要的组件了

###### 路由配置

第一步：安装 react-router-dom react-router-config

yarn add react-router-dom

yarn add react-router-config

安装 @loadable/component 实现路由组件按需加载

yarn add @loadable/component

安装完成，一切准备就绪，下面我们就开始配置路由

第二步：在src文件夹下新建一个路由文件夹，用于放置路由的配置文件，命名 router ；在 router 里新建一个 router.js

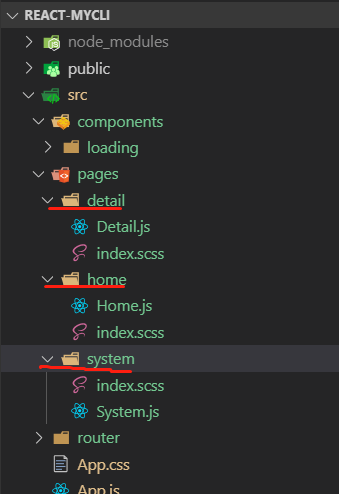
引入所需要使用的依赖包

import React from "react";

// loadable 用来实现路由组件的按需加载

import loadable from "@loadable/component";

创建组件并引入，如下所示：

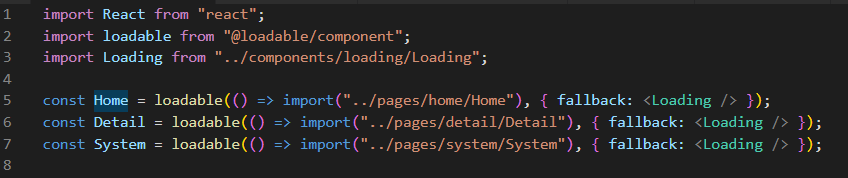


const Home = loadable(() => import("../pages/home/Home"));

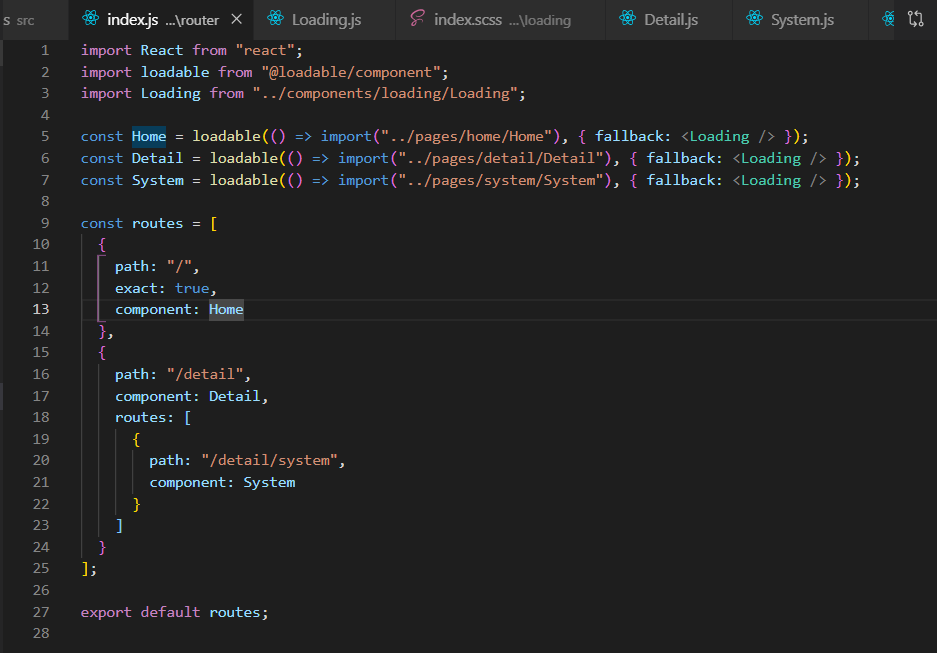
const Detail = loadable(() => import("../pages/detail/Detail"));

const System = loadable(() => import("../pages/system/System"));

创建Loading 组件，引入，用于组件加载前loading显示，增加用户体验，如下所示，至于为什么这样使用，请看 [@loadable/component 文档](https://www.npmjs.com/package/@loadable/component)



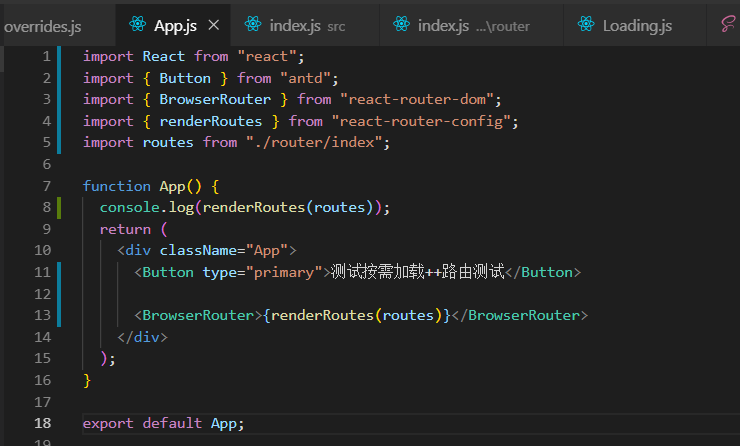
接下来就来配置路由数组吧，最终router/index.js 配置内容如下所示：



到这一步路由基本上配置完成了，下面来看看如何使用吧

首先进入 App.js 引入 react-router-dom react-router-config

其次引入路由配置文件 router/index.js 具体实现代码如下所示：

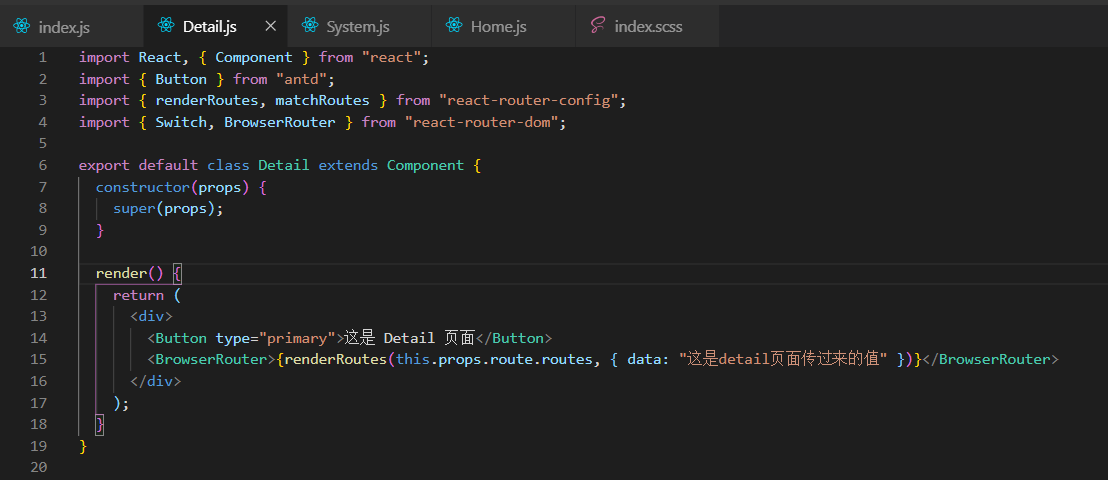


renderRoutes 是react-router-config 提供的api ，具体使用方法请参考 [react-router-config 文档](https://www.npmjs.com/package/react-router-config)

BrowserRouter 是 react-router-dom 提供的 api ，具体使用方法请参考[react-router-dom 文档](https://github.com/ReactTraining/react-router/tree/master/packages/react-router-dom)

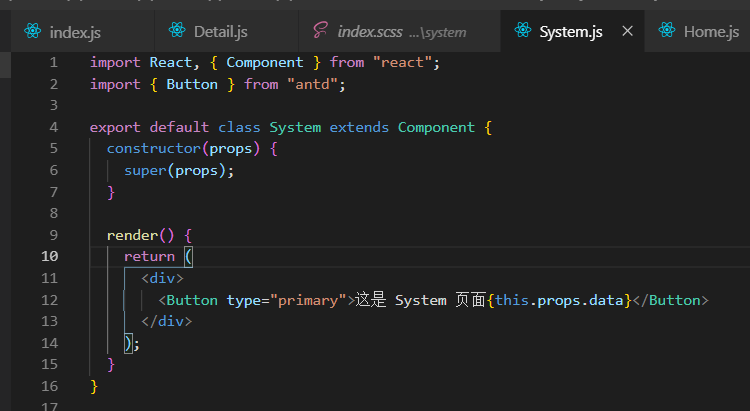
到这一步，第一层理由已经可以实现了，浏览器里输入对应的路由，即可显示对应的组件；下面来说说嵌套路由（子路由）如何实现：

从上面路由配置文件 router/index.js 文件中可以看出，/detail 下面存在子路由 /detail/system 要想实现其实很简单，只需要在Detail 组件中重新调用一下 renderRoutes 方法即可，当然在调用之前，还是需要取引用一下 react-router-config 和 react-router-dom ,子路由的获取，通过 this.props.route 获取，详细用法参考以下代码：



renderRoutes 第二个参数是一个对象，作用是向子组件中传值用，

子组件获取穿过来值得方法，只需要通过 this.props.属性 即可获取；代码实现如下：



到这里，路由的配置就已经完成了。