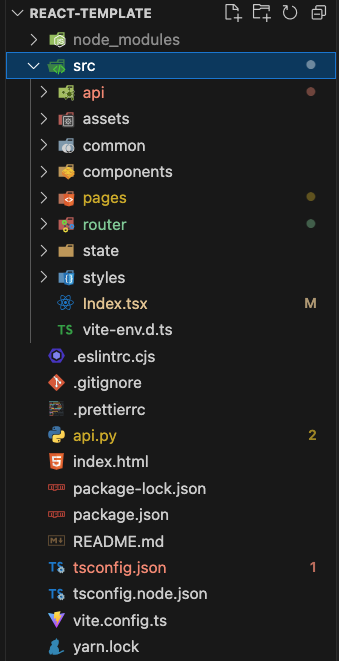
**React项目初始化模板**

**技术栈**

React18、Ts、Vite、react-use、recoil、antd等技术

**目录结构**



**index.html**

|  |
| --- |
| JavaScript <!DOCTYPE html> <html lang="en">  <head>  <meta charset="UTF-8" />  <link rel="icon" type="image/svg+xml" href="/vite.svg" />  <meta name="viewport" content="width=device-width, initial-scale=1.0" />  <title>React项目初始化模板</title>  </head>  <body>  <div id="root"></div>  <script type="module" src="/src/index.tsx"></script>  </body> </html> |

**vite.config.ts**

|  |
| --- |
| JavaScript import { defineConfig } from 'vite'; import react from '@vitejs/plugin-react';  // https://vitejs.dev/config/ export default defineConfig({  plugins: [react()],  server: {  proxy: {  '/dev-api': {  target: 'http://127.0.0.1:8002/',  changeOrigin: true,  },  '/api': {  target: 'https://bbs.hupu.com/',  changeOrigin: true,  },  '/mcp': {  target: 'https://ug.baidu.com/',  changeOrigin: true,  },  },  },  // 配置开发服务器中的预览功能、允许你在本地开发服务器中查看静态文件  preview: {  proxy: {  '/api': {  target: 'http://127.0.0.1:4523/m1/2899312-0-default/',  changeOrigin: true,  },  },  }, }); |

**api.py**

本地启动mock数据接口的服务

**src**

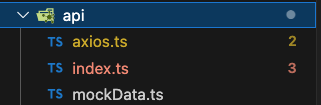
**index.tsx**

|  |
| --- |
| JavaScript import ReactDOM from 'react-dom/client'; import Router from './Router.tsx'; import 'antd/dist/reset.css'; import { RecoilRoot } from 'recoil';  ReactDOM.createRoot(document.getElementById('root') as HTMLElement).render(  <RecoilRoot>  <Router />  </RecoilRoot>, ); |

**RecoilRoot** 是状态管理库，类似redux

详细介绍：https://zhuanlan.zhihu.com/p/572425602?utm\_id=0

**api**



**axios.ts**

axios.ts

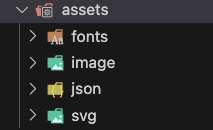
|  |
| --- |
| JavaScript import { message } from 'antd'; import axios from 'axios';  axios.interceptors.request.use(config => {  if (config.method === 'get') {  if (!config.params) {  config.params = {};  }  config.params.\_t = Date.now();  }  return config; });  axios.interceptors.response.use(  response => {  const { code, msg } = response.data;  if (code !== 200 && code !== 0) {  message.error(`code：${msg}`);  return Promise.reject(msg);  }  return response.data; /\* \*/  },  error => {  console.log(error);  return Promise.reject(error);  }, ); |

**index.ts**

前端请求接口函数配置

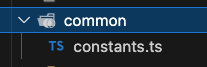
|  |
| --- |
| JavaScript import './axios'; import axios from 'axios';  interface Response<T> {  code: 200 | 500;  msg: string;  data: T; }  export interface Ratio {  [key: string]: number; }  export interface DevDataInfo {  name: string,  sex: string,  age: number, }  export interface OtherInfo {  otherProps: Ratio[]; }  export const api = {  // 本地接口假数据测试  getdevData:async (url?: string): Promise<Response<DevDataInfo>> => {  const { data } = await axios.get('/dev-api/code', { params: { url } });  return data;  },   // 调接口方法测试  getTestData:async (url: string): Promise<{}> => {  const { data } = await axios.get('/api/v2/nav', { params: { url } });  return data;  },  postTestData: async (params: any): Promise<{}> => {  const { data } = await axios.post('/mcp/pc/pcsearch', params);  return data;  }, }; |

**assets**



fonts：字体； image：图片； json：静态数据； svg：svg的dom代码

**common**

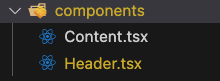


**constants.ts**

存放公共的各类静态自定义变量

|  |
| --- |
| JavaScript const fontSize = window.screen.width >= 3840 ? 24 : 12; export const labelTextStyle = {  fontSize,  fontFamily: 'PingFang SC',  color: '#CAC056',  lineHeight: 35, }; |

**components**

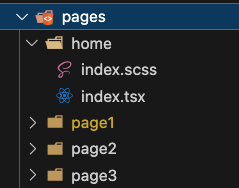


存放前端公共组件，可以是ui组件、也可以是业务组件。

**Content.tsx**

|  |
| --- |
| JavaScript import React, { memo } from "react"; import "../styles/content.scss";  interface Props {  children?: React.ReactNode; } const Content: React.FC<Props> = memo((props) => {  return <div className='content'>{props.children}</div>; }); export default Content; |

**pages**



存放前端页面的文件

**router/index.tsx**

配置页面路由

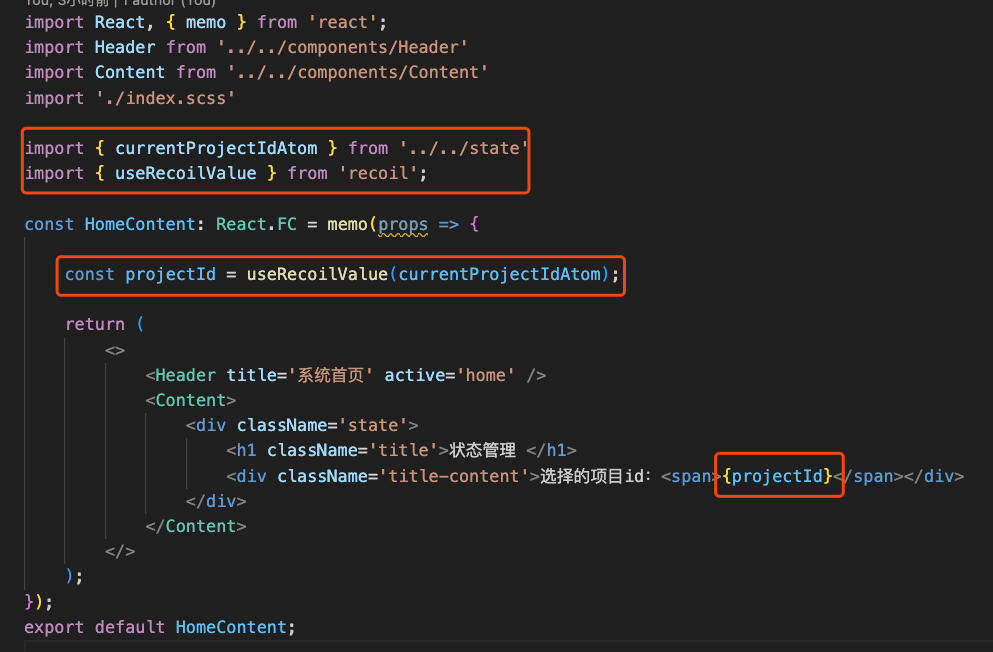
|  |
| --- |
| JavaScript import './Router.scss'; import './reset.scss'; import { RouterProvider, createBrowserRouter } from 'react-router-dom'; import Home from './pages/home'; import Page1 from './pages/page1'; import Page2 from './pages/page2';  function Router() {  const router = createBrowserRouter([  {  path: '/',  element: <Home />,  },  {  path: '/page1',  element: <Page1 />,  },  {  path: '/page2',  element: <Page2 />,  },  ]);  return <RouterProvider router={router} />; }  export default Router; |

**state/index.tsx**

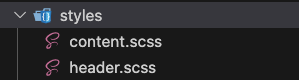
状态管理数据

|  |
| --- |
| JavaScript import { atom } from 'recoil';  export const currentProjectIdAtom = atom<string | null>({  key: 'currentProjectIdAtom',  default: null, }); |

用法：

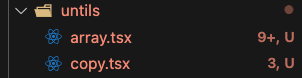


**styles**



公共样式

**untils**



公共工具