

# react项目常用知识点

## 项目搭建

基于脚手架搭建项目基本结构步骤：

1. 使用 React CLI 搭建项目： `npx create-react-app project-xx`
2. 进入项目根目录： `cd project-xx`
3. 启动项目： `yarn start`
4. 调整项目目录结构：
5. 修改页面标题

js 复制代码

```
/src
/  assets      项目资源文件，比如，图片 等
/  components  通用组件
/  pages       页面组件
/  store       Redux 状态仓库
/  utils       工具，比如，token、axios 的封装等
App.css       根组件样式文件
App.js        根组件
index.css     全局样式
index.js      项目入口
```

## 使用CSS预编译器 - SASS

SASS是一种预编译的 CSS，作用类似于 Less、stylus。由于 React 中内置了处理 SASS 的配置，所以，在 CRA 创建的项目中，可以直接使用 SASS 来写样式。

SASS 支持两种后缀，分别是： `.sass` 和 `.scss`。区别如下：

1. `.sass` 是一种简化语法形式（**用缩进代替 {}**、**用换行代替;**）

js 复制代码

```
.button
  display: inline-flex
  border: none

  &:hover
    cursor: pointer
```

## 2. `.scss` 是传统的语法形式【推荐使用】

js 复制代码

```
.button {
  display: inline-flex;
  border: none;

  &:hover {
    cursor: pointer;
  }
}
```

使用步骤：

1. 安装解析 sass 的包： `yarn add sass`
2. 将全局样式文件 `index.css` 修改为 `index.scss`，并修改 `index.js` 中导入的样式文件后缀
3. 将 `App.css` 修改为 `App.scss`，并修改 `App.js` 中导入样式文件的后缀

## 配置基础路由

配置基础路由步骤：

1. 安装路由： `yarn add react-router-dom@5.3.0`
2. 在 `pages` 目录中创建两个文件夹：Login、Layout、NotFound
3. 分别在三个目录中创建 `index.js` 文件，并创建一个简单的组件后导出
4. 在 `App` 组件中，导入路由组件以及 3 个页面组件
5. 配置 Login、Layout、NotFound 的路由规则

在 `App.js` 页面中配置基础路由代码如下：

js 复制代码

```
// 导入路由
import { BrowserRouter as Router, Route, Switch } from 'react-router-dom'
```

```
// 导入页面组件
import Login from './pages/Login'
import Layout from './pages/Layout'
import NotFound from './pages/NotFound'

// 配置路由规则
function App() {
  return (
    <Router>
      <div className="App">
        <Switch>
          <Route path="/home" component={Layout}></Route>
          <Route path="/login" component={Login}></Route>

          <Route>
            <NotFound />
          </Route>
        </Switch>
      </div>
    </Router>
  )
}

export default App
```

## 默认展示首页内容

在项目中，我们希望能够在打开页面时就展示首页内容。实现这个需求的思路是匹配默认路由，进行重定向。为了实现该功能，需要用到路由的两个内容：

1. Route 组件的 render 属性
2. Redirect 组件实现路由跳转

原来的做法是使用 component 属性，需要指定要渲染的组件，在使用的时候，必须先创建一个组件，然后，设置为 component 属性的值，才能渲染。

js 复制代码

```
<Route path="/" component={Login} />
```

现在我们可以使用 Route 的 render 属性，用来内联渲染任意内容，不需要额外的创建一个组件。render 属性的值是函数，通过函数的返回值来指定要渲染的内容。

js 复制代码

```
<Route path="/" render={() => <div>内联在 Route 标签上的内容</div>} />
```

路由重定向。

js 复制代码

```
import { Redirect } from 'react-router-dom'  
// 路由重定向组件：  
<Redirect to="/login" />
```

默认展示首页内容实现步骤：

1. 在 App.js 中添加一个新的 Route，用来匹配默认路由
2. 为 Route 组件添加 `render` 属性，用来渲染自定义内容
3. 在 render 中，渲染 Redirect 实现路由重定向

js 复制代码

```
import { Redirect } from 'react-router-dom'  
  
<Route exact path="/" render={() => <Redirect to="/home" />} />
```