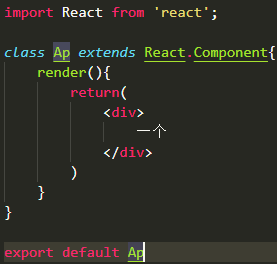
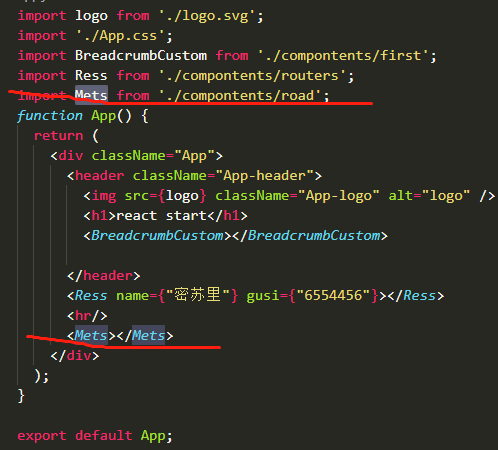
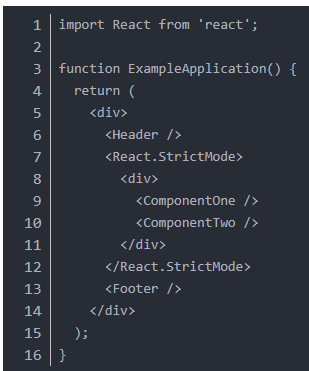
**0.react组件引入方式**



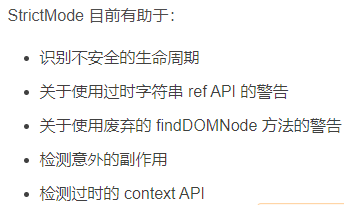
**暴露出Ap，在其他页面用important引入该组件，用标签把组件名写入**



**1.React.StrictMode**

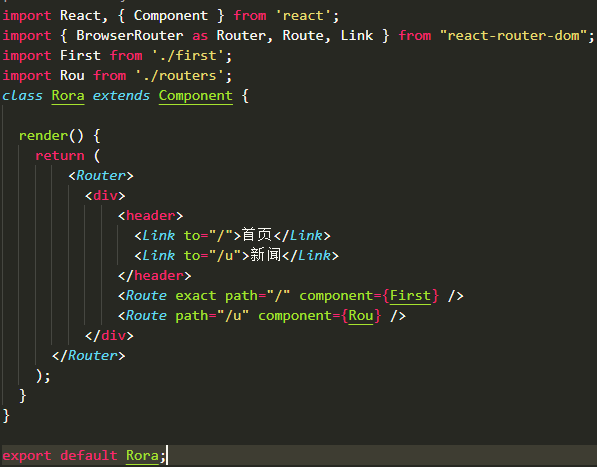
不会对 Header 和 Footer 组件运行严格模式检查。但是，ComponentOne 和 ComponentTwo 以及它们的所有后代元素都将进行检查

优点：



**2.路由设定（基本版）**

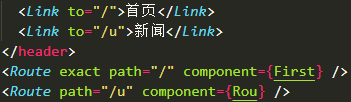
**安装步骤：**



**1.yarn add react-router-dom 或者 npm install react-router-dom 安装指令module包**

**2.引入 import {BrowserRouter as Router,Route,link} from 'react-router-dom';**

**3.引入要跳转的页面**

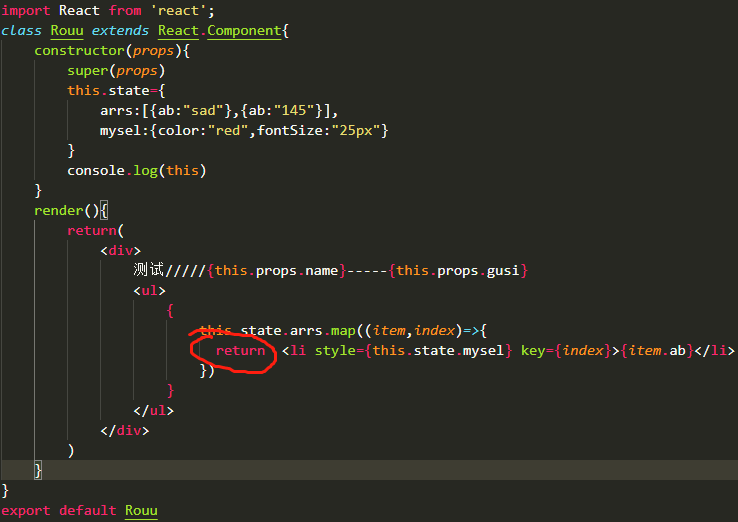
**4.** **link将在页面中渲染成a标签，component={引入的组件名}**

**5.exact**

**exact是Route下的一条属性，react路由会匹配所有匹配到的路由组价，exact能够使得路由的匹配更严格一些。为了解决route的唯一渲染，它是为了保证路由只渲染一个路径。**

**exact的值为bool型，为true是表示严格匹配，为false时为正常匹配（一般加在path=“/” 的标签上）**

**3.循环**



**脚手架中写循环要加上return 否则无法实现**

**4.抽离路由，进行路由配置（脚手架版）**

**参考** [**https://blog.csdn.net/qq\_33455771/article/details/83375893**](https://blog.csdn.net/qq_33455771/article/details/83375893)

**获取当前的路由 this.props.match**

1. **安装路由包**

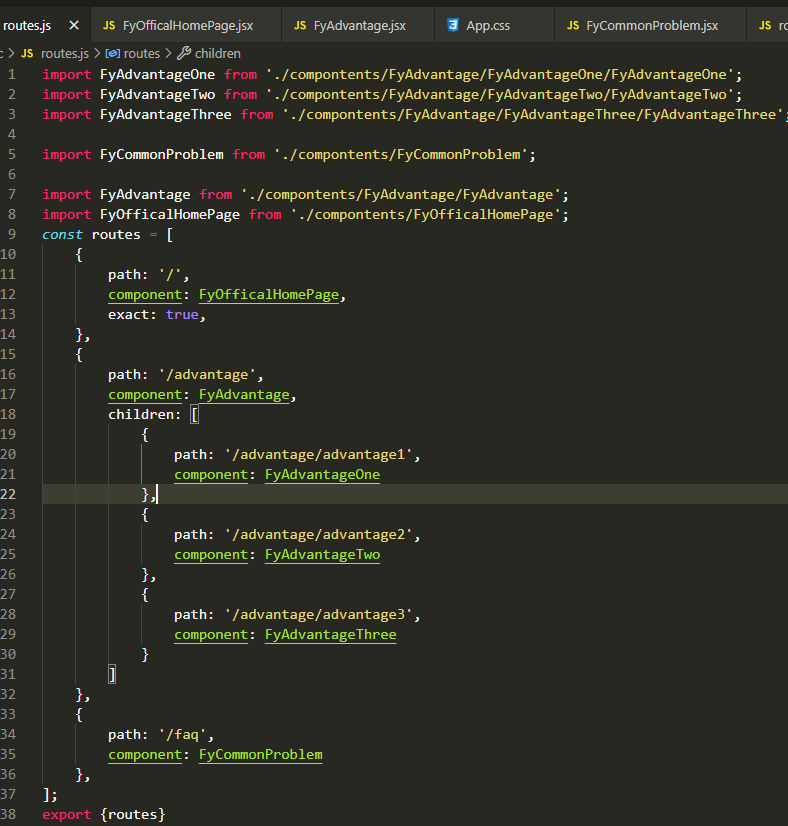
**（1）npm install --save-dev react-router**

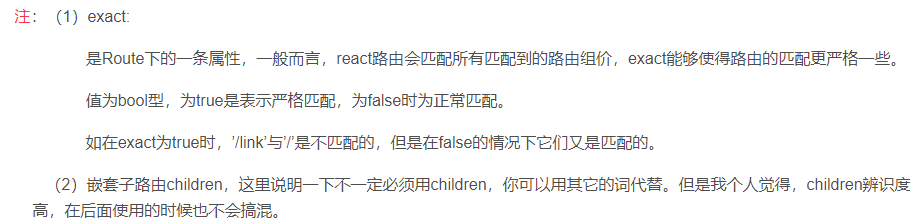
**（2）npm install --save-dev react-router-dom**

**（3）npm install --save react-router-config**

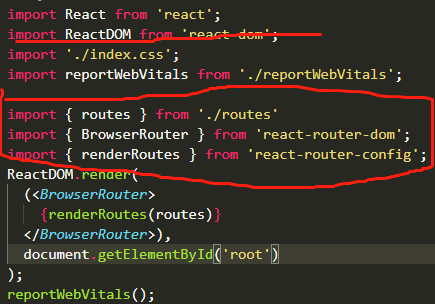
**②配置路由**

1. **建一个route.js文件**



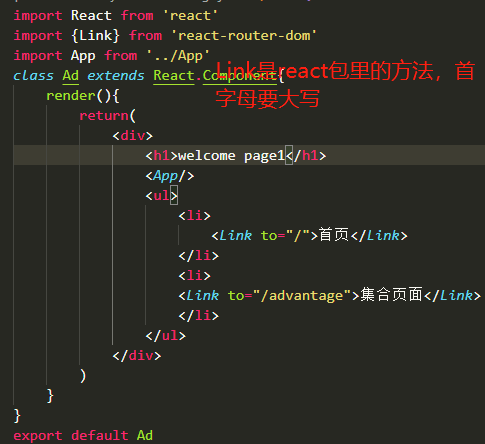
**exact类似vue路由重定向**

1. **在index.js中引入路由模块**



**Routes是routes中导出的配置**

1. **在组件中运用**



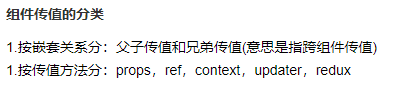
**（4）子路由界面**



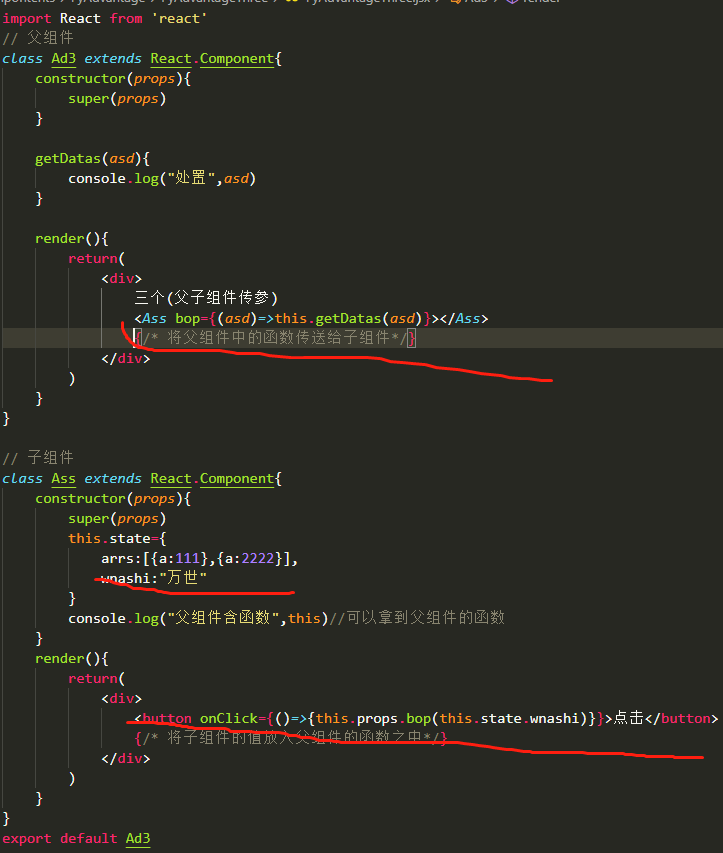
**{renderRoutes(route.children)}是子页面的入口，子页面是在原来的页面的基础加载的，**

**5组件传参**

1. **传参分类：**



1. **子传父 父组件通过props向子组件传入一个方法，子组件在通过点击事件调用该方法，将数据以参数的形式传给父组件，父组件可以在该方法中对传入的数据进行处理；**



**特注：**



1. **路由传参**