

Router

本节课内容

React Router

理解了路由基本机制以后，也不需要重复造轮子，我们可以直接使用 React Router 库
<https://reacttraining.com/react-router/>

React Router 提供了多种不同环境下的路由库

- web
- native

基于 Web 的 React Router

基于 web 的 React Router 为：react-router-dom

安装

```
npm i -S react-router-dom
```

组件

传递 props

```
<Route exact path="/" component={Home}>
```

如果 Route 使用的是 component 来指定组件，那么不能使用 props

Route : render

```
<Route exact path="/" render={() => <Home items={this.state.items} />} />
```

通过 render 属性来指定渲染函数，render 属性值是一个函数，当路由匹配的时候指定该函数进行渲染

NavLink 组件

NavLink 与 Link 类似，但是它提供了两个特殊属性用来处理页面导航

activeStyle

当当前 URL 与 NavLink 中的 to 匹配的时候，激活 activeStyle 中的样式

activeClassName

与 activeStyle 类似，但是激活的是 className

isActive

默认情况下，匹配的是 URL 与 to 的设置，通过 isActive 可以自定义激活逻辑，isActive 是一个函数，返回布尔值

Switch 组件

该组件只会渲染首个被匹配的组件

Redirect 组件

to

设置跳转的 URL

路由传参

withRouter 组件

如果一个组件不是路由绑定组件，那么该组件的 props 中是没有路由相关对象的，虽然我们可以通过传参的方式传入，但是如果结构复杂，这样做会特别的繁琐。幸好，我们可以通过 withRouter 方法来注入路由对象

hooks

- useHistory
- useLocation
- useParams
- useRouteMatch

下节课内容

- Redux
- React-Redux

练习

- 基于给定数据 和 视图 React-Router 完成翻页导航效果
- 该效果具备以下功能
 - 每页显示5条数据
 - 点击翻页按钮修改URL
 - URL 变化之后获取当前页码，根据页码获取数据显示
 - 重定向至 404 页面