



黑马程序员™
www.itheima.com

传智播客旗下
高端IT教育品牌

React 路由基础

目录 Contents

◆ React路由介绍

◆ 路由的基本使用

◆ 路由的执行过程

◆ 编程式导航

◆ 默认路由

◆ 匹配模式

1. React路由介绍

现代的前端应用大多都是 SPA (单页应用程序)，也就是只有一个 HTML 页面的应用程序。因为它的用户体验更好、对服务器的压力更小，所以更受欢迎。为了有效的使用单个页面来管理原来多页面的功能，前端路由应运而生。

- 前端路由的功能：让用户从一个视图（页面）导航到另一个视图（页面）
- 前端路由是一套映射规则，在React中，是 URL路径 与 组件 的对应关系
- 使用React路由简单来说，就是配置 路径和组件（配对）

目录 Contents

- ◆ React路由介绍
- ◆ 路由的基本使用
- ◆ 路由的执行过程
- ◆ 程式化导航
- ◆ 默认路由
- ◆ 匹配模式

2. 路由的基本使用

2.1 使用步骤

1. 安装: yarn add react-router-dom
2. 导入路由的三个核心组件: Router / Route / Link

```
import { BrowserRouter as Router, Route, Link } from 'react-router-dom'
```

3. 使用 Router 组件包裹整个应用 (重要)

```
<Router>  
  <div className="App">  
    // ... 省略页面内容  
  </div>  
</Router>
```

2. 路由的基本使用

2.1 使用步骤

4. 使用 **Link 组件**作为导航菜单（路由入口）

```
<Link to="/first">页面一</Link>
```

5. 使用 **Route 组件**配置路由规则 and 要展示的组件（路由出口）

```
const First = () => <p>页面一的页面内容</p>

<Router>
  <div className="App">
    <Link to="/first">页面一</Link>
    <Route path="/first" component={First}></Route>
  </div>
</Router>
```

2. 路由的基本使用

2.2 常用组件说明

- Router 组件：包裹整个应用，一个 React 应用只需要使用一次
- 两种常用 Router：HashRouter 和 BrowserRouter
- HashRouter：使用 URL 的哈希值实现 (localhost:3000/#/first)
- (推荐) BrowserRouter：使用 H5 的 history API 实现 (localhost:3000/first)

2. 路由的基本使用

2.2 常用组件说明

- Link 组件：用于指定导航链接（a 标签）

```
// to属性：浏览器地址栏中的pathname（location.pathname）  
<Link to="/first">页面一</Link>
```

- Route 组件：指定路由展示组件相关信息

```
// path属性：路由规则  
// component属性：展示的组件  
// Route组件写在哪，渲染出来的组件就展示在哪  
<Route path="/first" component={First}></Route>
```

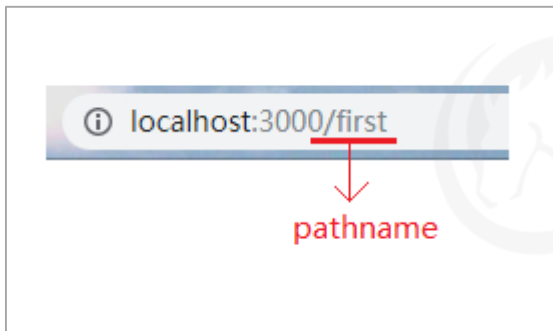

目录

Contents

- ◆ React路由介绍
- ◆ 路由的基本使用
- ◆ 路由的执行过程
- ◆ 程式化导航
- ◆ 默认路由
- ◆ 匹配模式

3. 路由的执行过程

1. 点击 Link 组件 (a 标签), 修改了浏览器地址栏中的 url。
2. React 路由监听到地址栏 url 的变化。
3. React 路由内部遍历所有 Route 组件, 使用路由规则 (path) 与 pathname 进行匹配。
4. 当路由规则 (path) 能够匹配地址栏中的 pathname 时, 就展示该 Route 组件的内容。



目录 Contents

- ◆ React路由介绍
- ◆ 路由的基本使用
- ◆ 路由的执行过程
- ◆ 程式化导航
- ◆ 默认路由
- ◆ 匹配模式

4. 程式导航

- 场景：点击登录按钮，登录成功后，通过代码跳转到后台首页，如何实现？
- 程式导航：通过 JS 代码来实现页面跳转
- history 是 React 路由提供的，用于获取浏览器历史记录的相关信息
- push(path)：跳转到某个页面，参数 path 表示要跳转的路径
- go(n)：前进或后退到某个页面，参数 n 表示前进或后退页面数量（比如：-1 表示后退到上一页）

```
class Login extends Component {  
  handleLogin = () => {  
    // ...  
    this.props.history.push('/home')  
  }  
  render() {...省略其他代码}  
}
```

目录

Contents

- ◆ React路由介绍
- ◆ 路由的基本使用
- ◆ 路由的执行过程
- ◆ 程式化导航
- ◆ 默认路由
- ◆ 匹配模式

5. 默认路由

- 问题：现在的路由都是点击导航菜单后展示的，如何在进入页面的时候就展示呢？
- 默认路由：表示进入页面时就会匹配的路由
- 默认路由path为：/

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

目录 Contents

- ◆ React路由介绍
- ◆ 路由的基本使用
- ◆ 路由的执行过程
- ◆ 编程式导航
- ◆ 默认路由
- ◆ 匹配模式

6. 匹配模式

6.1 模糊匹配模式

- 问题：当 Link 组件的 to 属性值为 “/login” 时，为什么 默认路由 也被匹配成功？
- 默认情况下，React 路由是模糊匹配模式
- 模糊匹配规则：只要 pathname 以 path 开头就会匹配成功

```
<Link to="/login">登录页面</Link>  
<Route path="/" component={Home} /> 匹配成功
```

```
// path 代表Route组件的path属性  
// pathname 代表Link组件的to属性（也就是 location.pathname）
```

path	能够匹配的pathname
/	所有 pathname
/first	/first 或 /first/a 或 /first/a/b/...

6. 匹配模式

6.2 精确匹配

- 问题：默认路由任何情况下都会展示，如何避免这种问题？
- 给 Route 组件添加 `exact` 属性，让其变为精确匹配模式
- 精确匹配：只有当 `path` 和 `pathname` 完全匹配时才会展示该路由

// 此时，该组件只能匹配 `pathname="/"` 这一种情况

```
<Route exact path="/" component=... />
```

推荐：给默认路由添加 `exact` 属性。

总结

React 路由基础

1. React 路由可以有效的管理多个视图（组件）实现 SPA
2. Router组件包裹整个应用，只需要使用一次
3. Link组件是入口，Route组件是出口
4. 通过 props.history 实现程式化导航
5. 默认模糊匹配，添加 exact 变精确匹配
6. React 路由的一切都是组件，可以像思考组件一样思考路由



黑马程序员

www.itheima.com

传智播客旗下高端IT教育品牌