

# React 程序开发

## --- React 路由



河北师范大学软件学院  
Software College of Hebei Normal University

# 内容提纲

---

➤ **路由配置**

➤ **动态路由**



# React 路由

---

- React 路由

- react-router 是浏览器和原生应用的通用部分
- react-router-native 是用于原生应用的
- react-router-dom 是用于浏览器的

- 安装

- `npm install --save react-router-dom`

- 引入

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



# 路由配置

```
import { BrowserRouter as Router,Route,Link} from 'react-router-dom';
.....
class Hello extends React.Component {
  render() {
    return (
      <Router>
        <div>
          <Route exact path="/" component={Home}/>
          <Route path="/about" component={About}/>
        </div>
      </Router>
    );
  }
}
```

# 路由配置

---

- **BrowserRouter VS HashRouter**

- 都是路由的基本组件，需将其放在最外层，选其一使用
- 内部只能含有一个子节点
- BrowserRouter 的 URL 是指向真实 URL 的资源路径，页面和浏览器的 history 保持一致
- HashRouter 使用 URL 中的 hash (#) 部分去创建路由

- **<Route />**

- 匹配路径、挂载组件



# 路由配置

---

- `<Route />` 续
  - `<Route exact path='/home' component={ Home }/>`
  - `exact` : 严格匹配
  - `path` : 字符串类型, 用来匹配url
  - `component` : 值是一个组件, 在 `path` 匹配成功后挂载这个组件
  - `render` : 一个返回组件的方法
  - `children` : 一个返回组件的方法

# 路由配置

---

- `<Switch > ... </Switch>`
  - 只会挂载与 `<Route>` 路径匹配成功的第一个
  - 子节点只能是 `<Route>` 或 `<Redirect />`

```
<Switch>
```

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

```
  <Route path="/about" component={ About }/>
```

```
  <Route path="/:user" component={ User }/>
```

```
  <Route component={ NoMatch }/>
```

```
</Switch>
```



# 路由配置

---

- `<Link to = "... " > ... </Link>`

**to : string**

```
<Link to="/home">Home</Link>
```

**to : object**

```
<Link to={ {  
  pathname: '/courses',  
  search: '?sort=name',  
  hash: '#the-hash',  
  state:{id : 1}
```

**// 传到组件的数据, 通过 `this.props.location.state` 获取**

```
} } > Courses </Link>
```





# 路由配置

---

- `<NavLink to = "... " > ... </NavLink>`
  - NavLink 是 Link 的子类
  - activeClassName: string, 被选中时添加的类名
  - activeStyle: object, 被选中时添加的行内样式
  - exact: boolean, 严格匹配

# 路由配置

- `<Redirect to = "... " >`
  - 当被挂载时，将路由重定向为 to 属性指定的地址
  - `<Route path="/home" render={() => <Redirect to="/other " /> }/>`
  - from 和 exact 两个属性只能用在 `<Switch>` 组件下的 `<Redirect />`

```
<Switch>  
  <Redirect from = '/old' to = '/new' />  
  <Route path='/new' component={ New }/>  
</Switch>
```

# 内容提纲

---

➤ 路由配置

➤ 动态路由



# 动态路由

---

- 路由配置

- `<Route path='/content/:id' component={ Content }/>`
- `<Link to={ '/content/1' }>{ item.title }</Link>`

- 数据接收

- match 对象
  - 一个包含 `<Route path= " " >` 匹配路径参信息的对象
  - params
  - url、path、isExact

# 路由 get 传值

---

- 路由配置

- `<Route path='/content' component={ Content }/>`
- `<Link to={ `/content?id=1` }>{item.title}</Link>`

- 数据接收

- 数据存放在 `this.props.location.search`
- 通过 URL 模块解析
  - `npm install url --save`
  - `url.parse(urlStr : string,parseQueryString : true)`

The background of the slide is decorated with various abstract shapes in shades of green and yellow. These shapes, which include circles, ovals, and teardrops, are scattered across the top and right sides of the slide, creating a modern and organic feel.

# Thank You