



什么是单页面应用 SPA

第一堂设计课 Material Design

第二章：安装 Meteor，学会使用 React 和 Material-ui

创建一个 Meteor 应用

在 Meteor 应用中使用 React

添加路由

SPA 应用开发的一个挑战就是，怎么样让单页面看起来有“多个页面”的效果。这会涉及到“路由控制”（route）的问题。本节将通过在 React 社区非常流行的 **react-router** 路由库来实现路由功能。

安装 react-router

首先安装 npm 包，[参考](#)，在应用根目录下运行命令：

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

查看更改：[安装 react-router](#)

添加三个新组件

在使用 react-router 之前，先在 imports/ui 目录下添加三个组件文件：SignUp.jsx、LogIn.jsx、Home.jsx，以 Home.jsx 文件为例：

```
import React, { Component } from 'react';
import { Link } from 'react-router';

class Home extends Component {
  render() {
    return (
      <div>
        <div>
          <Link to='/'>Home</Link>
          <Link to='/signup'>Sign Up</Link>
          <Link to='/login'>Log In</Link>
        </div>
        This is home page.
      </div>
    );
  }
}

export default Home;
```

通过 React Router 的 [Link](#) 组件创建了三个链接。

查看更改：[添加三个新组件](#)

定义简单的路由

根据官方文档中给出的[文件结构范例](#)一节内容，路由文件应该放到

`imports/startup/client/` 目录下，执行下面命令创建一系列文件：

```
mkdir imports/startup && mkdir imports/startup/client
cd imports/startup/client && touch routes.jsx
```

接下来，添加一些代码到 `routes.jsx` 文件中：

```
import React from 'react';
import { Router, Route, browserHistory } from 'react-router'

import SignUp from '../../ui/SignUp.jsx';
import LogIn from '../../ui/LogIn.jsx';
import Home from '../../ui/Home.jsx';

export const renderRoutes = () => (
  <Router history={browserHistory}>
    <Route path="/" component={Home} />
    <Route path="/signup" component={SignUp} />
    <Route path="/login" component={LogIn} />
  </Router>
);
```

注意，这里的各个 `path` 的值要和前面的 `<Link to='/signup'>` 对应的各个 `to=` 值保持一致。

查看更改：[定义简单的路由](#)

首先从 npm 包 `react-router` 中导入所需的功能模块 `Router`、`Route`、`browserHistory`，这里采用是 ES6 模块的[命名导出](#)方式所对应的导入语法格式，模块名字被花括号包裹起来。

然后导入之前新编写的 `SignUp`、`LogIn` 和 `Home` 组件，这里采用的是 ES6 模块的[默认导出](#)方式所对应的导入语法格式，名字没有包裹在花括号里面。

接下来利用从 `react-router` 导入的三个功能模块来设置应用的路由功能。通过 `Route` 组件，把页面的 URL 和 要渲染的组件对应起来，并把它们做为子组件包裹到 `Router` 组件

中。

着重介绍一下 browserHistory

注意上面代码中出现了 `browserHistory` 的字样。那么这里 `History` 也就是历史，是干神马的呢？

到 [MDN 上的 HTML5 简介页面](#)，可以看到专门有 **History API** 接口这一部分。可见 `history` 是浏览器内置的功能。这里我们要特别关注的是 `history.pushState` 接口。

可以打开我们 `chrome` 的开发者工具，选中 `js console`，来运行一下 [mdn 上给的例子](#)

先来执行：

```
var stateObj = { foo: "bar" };  
history.pushState(stateObj, "page 2", "/peter.html");
```

这样可以看到，浏览器地址栏中，路径变成了 `/peter.html` 然后执行

```
history.pushState(stateObj, "page 2", "/billie.html");
```

这时候路径又变成了 `/billie.html`。

由此可见，`history.pushState` 接口是可以自由操作浏览器 URL 的，这个也就是 `react-router` 可以工作的底层机制保障。同时，通过这个接口来改变的 URL 是记录在浏览器历史中的（这也就是 `history` 这个 API 名字的由来了），表象是，点浏览器的“前进”和“后退”按钮，是可以在 `/peter.html` 和 `/billie.html` 之间切换的。

`browserHistory` 就是 `React Router` 对 `History API` 进行的封装。更多说明可以参考官方文档 [Histories](#)。

ES6 知识小科普

上面的代码中用到了 ES6 [胖箭头函数](#)的写法，与之等价的 ES5 函数写法是：

```
export const renderRoutes = function(){
  return (
    ...
  );
}
```

渲染路由组件

打开 client/main.jsx 文件，修改如下：

```
...
import { renderRoutes } from '../imports/startup/client/routes.jsx';

Meteor.startup(() => {
  injectTapEventPlugin();
  render(renderRoutes(), document.getElementById('app-container'));
});
```

查看更改：[渲染路由组件](#)

到底渲染哪个组件？这个由 React Router 负责通知 `render()` 方法当前应该把哪个组件挂载到 `app-container` 节点上。

结语

访问 localhost:3000 页面，点击页面中的三个链接，就可以在不同页面之间进行切换了。这样就让一个单页面应用看起来有了多个页面。

欢迎添加 Peter 的微信：happypeter1983

