



第三章：路由控制

添加路由

嵌套路由

美化导航栏

导航栏状态控制

第四章：Inline Style 专题

内联样式中使用媒体查询

本节主要演示一下如何借助 Radium 提供的功能在 React 组件的内联样式中使用媒体查询，实现组件的响应式布局，还会介绍一下组件中如何使用全局样式。

填充 Home 组件的内容

为了更好的演示 Radium 的媒体查询功能，我们先打扮一下 Home 组件，添加一张图片 home-bg.jpg 和一行标语 chat room demo。需要说明一下 Meteor 应用中静态资产（如图片等）存放在应用根目录下的 ./public 目录中，如本节图片存放位置是 ./public/images 目录，当使用图片的时候，图片路径就直接写成 /images/home-bg.jpg。另外，还添加了很多样式代码：

查看更改：[美化 Home 组件](#)

同样，我们把 Home 组件用 Radium() 包裹起来。这里着重看一下关于媒体查询的样式代码：

```
slogan: {
  ...
  '@media (min-width: 600px)': {
    fontSize: '50px',
    paddingBottom: '100px',
    paddingTop: '100px'
  }
}
```

意思是说在大屏幕设备上，chat room demo 这行标语的字体会变大且上下内边距也变宽，实现响应式布局。不幸的是，这会儿到浏览器中访问页面，一片空白，浏览器控制台报出错误信息：

```
Uncaught Error: To use plugins requiring `addCSS` (e.g. keyframes, medi
```

字面意思是“请把你的应用包裹在 StyleRoot 组件中，才能使用 media queries 插件”，所以我们需要导入 Radium 提供的 StyleRoot 组件模块，在 App.jsx 文件开头添加一行代码：

```
import Radium, { StyleRoot } from 'radium';
```

然后再把 App 组件 render 方法中返回的 jsx 代码包裹在 <StyleRoot>  
</StyleRoot> 组件中。保存文件，再去访问应用首页，会看到一张美丽的背景图片，

还有三个英文单词。若调整浏览器窗口的大小，会发现这三个英文单词的大小会发生改变，说明我们添加的媒体查询样式已经生效了。

不过，要实现真正的响应式布局，让应用在手机上也能清楚地展示，还要在 `main.html` 文件的 `<head></head>` 元素中添加一行代码：

```
<meta name="viewport" content="width=device-width, initial-scale=1" />
```

查看更改：[媒体查询生效了](#)

到此，我们的应用也可以在手机上清晰地显示了。打开 Chrome 开发者工具，点击**设备**图标，体验一下吧。不过，导航栏中的 `Tabs` 组件是固定宽度的，因此不能完美地适应手机屏幕，下一节将实现导航栏的响应式布局。

欢迎添加 Peter 的微信：happypeter1983

冀ICP备15007992号-3