



第一章 课程简介

概述

第二章 编写 React 组件

展示课程封面

添加路由

美丽的心

现在已经在项目首页显示所有的课程封面了，接下来要完成的任务是为每门课程添加点赞功能。当点击点赞按钮的时候，会在课程封面上出现一个漂亮的心形图案，图案上方显示课程赞的个数，随后心形图案会自动消失。下面我们就一步步实现这个功能。

显示心形图案

修改 `Course.js` 组件文件，添加代码：

```
<div style={styles.imgWrap}>
  ...
  <span className="likes-heart">{course.likes}</span>
</div>
```

上面代码中，`span` 组件就是心形图案组件，样式由 CSS 文件 `./styles/style.css` 控制的，代码如下：

```
.likes-heart {
  background: url(http://o86bpj665.bkt.clouddn.com/redux-tower/heart.png);
  background-size: contain;
  font-size: 2rem;
  padding: 1rem;
  position: absolute;
  color: #ff4081;
  left: 50%;
  top: 50%;
  pointer-events: none;
}

/* animation */
.likes-heart {
  opacity: 0;
  transition: all 0.5s;
  transform: translateX(-50%) translateY(-50%) scale(5);
  display: block;
}
```

默认心形图案是不显示的，可以把不透明度设置为1，让图案显示出来：

```
.likes-heart {
  opacity: 1;
}
```

为了加载 CSS 文件 `./styles/style.css`，还需要修改 `index.js` 文件，添加一行代码：

```
import css from './styles/style.css';
```

注意：因为要完成心形图案的动画效果，用内联样式不好实现，所以采用了外置的 CSS 文件。

编写 CourseActions 组件

CourseActions 组件用来显示课程点赞按钮和评论按钮。新建一个 `components/CourseActions.js` 组件文件，添加如下代码：

```
import React, { Component } from 'react';
import { Link } from 'react-router';
import ActionThumbUp from 'material-ui/svg-icons/action/thumb-up';
import CommunicationComment from 'material-ui/svg-icons/communication/c';
import Radium from 'radium';

class CourseActions extends Component {
  getStyles() {
    return {
      root: {
        borderTop: '1px solid #e2e2e2',
        padding: '1.5rem',
        display: 'flex',
        flexDirection: 'column',
        '@media (min-width: 600px)': {
          flexDirection: 'row',
          alignItems: 'center',
          justifyContent: 'space-between'
        },
      },
    },
    name: {
      color: '#4c5765',
      fontSize: '1.7rem',
      marginBottom: '20px',
      textAlign: 'center',
    },
  }
}
```

```

    '@media (min-width: 600px)': {
      marginBottom: 0,
    }
  },
  btnWrap: {
    display: 'flex',
    justifyContent: 'space-between',
  },
  a: {
    display: 'block',
    marginLeft: '10px',
    textDecoration: 'none'
  },
  button: {
    backgroundColor: '#f2f4f6',
    color: '#4c5765',
    height: '36px',
    minWidth: '80px',
    border: 'none',
    padding: '3px 6px',
    display: 'flex',
    alignItems: 'center',
    justifyContent: 'center',
    ':hover': {
      cursor: 'pointer'
    },
    ':focus': {
      outline: 'none'
    },
  },
},
icon: {
  width: '18px',
  height: '18px',
  paddingRight: '6px'
}
};
}

```

```

render() {
  const { course } = this.props;
  let styles = this.getStyles();

```

```

    return (
      <div style={styles.root}>
        <div style={styles.name}>{course.name}</div>
        <div style={styles.btnWrap}>
          <button style={styles.button} onClick={this.props.increment}>
            <ActionThumbUp color='#4c5765' style={styles.icon} />
            <div>{this.props.likes}</div>
          </button>
          <Link to={`/view/${course.id}`} style={styles.a}>
            <button style={styles.button} key='2'>
              <CommunicationComment color='#4c5765' style={styles.icon}>
                <div>0</div>
            </button>
          </Link>
        </div>
      </div>
    );
  }
}

export default Radium(CourseActions);

```

我们用到了 Material-UI 的 **SVG Icon** 组件 `ActionThumbUp` 和 `CommunicationComment`，它们分别对应着谷歌的两个 **材料图标**，这两个图标用来美化 `button` 组件，并且给每个 `button` 组件添加了一个 `key` 属性，要不然 Radium 会报告错误信息：

Radium requires each element with interactive styles to have a unique k

实现点赞按钮的代码如下：

```

<button style={styles.button} onClick={this.props.increment} key='1'>..

```

当点击点赞按钮的时候会触发 `onClick` 事件处理函数 `this.props.increment`，这个 `increment` 属性由 `Course` 组件传入 `CourseActions` 组件。

编写 increment 方法

修改 `Course.js` 组件文件，先定义一个 `state` 变量 `likes`，添加代码：

```
constructor(props) {  
  super(props);  
  this.state = {  
    likes: this.props.course.likes  
  }  
}
```

上述代码把 `state` 变量 `likes` 初始化为课程已经获取的赞个数。

然后，编写 `increment` 方法让 `likes` 状态值加1：

```
increment() {  
  this.setState({likes: this.state.likes + 1})  
}
```

最后，在 `Course` 组件中导入 `CourseActions` 组件并使用起来：

```
import CourseActions from './CourseActions';  
  
<Card>  
  ...  
  <CourseActions course={course} increment={this.increment.bind(this)}>  
</Card>
```

`CourseActions` 组件包裹在课程卡片 `Card` 组件中，它有 `course`、`increment` 和 `likes` 三个属性。这样，当在 `CourseAction` 组件中点击点赞按钮之后，会执行 `Course` 组件中的 `increment` 方法，让 `Course` 组件中的 `likes` 状态值加1。

同时修改一下心形图案组件的代码，让其正确显示的课程所集赞的个数：

```
<span className="likes-heart">{this.state.likes}</span>
```

实现心形图案动态特效

心形图案由小慢慢变大，然后再慢慢消失，这种动画效果是由 React 提供的一个附加组件实现的，这个附加组件可以完成组件基本的 CSS 动画和过渡效果，其来自 npm 包 `react-addons-css-transition-group`。

修改 `Course.js` 组件文件，首先从 `react-addons-css-transition-group` 中导入这个附加组件，导入的组件名称可以任意定义，本项目使用的名字是

`CSSTransitionGroup`：

```
import CSSTransitionGroup from 'react-addons-css-transition-group';
```

得到 `CSSTransitionGroup` 组件之后，就可以在 `Course` 组件中用起来了：

```
<div style={styles.imgWrap}>
  ...
  <CSSTransitionGroup transitionName="like" transitionEnterTimeout={500
    <span key={this.state.likes} className="likes-heart">{this.state.li
  </CSSTransitionGroup>
</div>
```

把 `span` 组件用 `CSSTransitionGroup` 组件包裹起来实现动画效果，必须给 `span` 组件添加一个 `key` 属性，并且 `key` 属性值是可变的。当 `key` 值改变的时候，`span` 元素会出现两个 CSS class `like-enter` 和 `like-enter-active`，名字中的 `like` 字符串对应 `CSSTransitionGroup` 组件的 `transitionName` 属性值。通过这两个 CSS class 可以控制心形组件出现时的动画效果，对应的样式代码如下：

```
.likes-heart.like-enter {
  transition: all .2s;
  transform: translateX(-50%) translateY(-50%) scale(1);
  opacity: 1;
}
.likes-heart.like-enter.like-enter-active {
  transform: translateX(-50%) translateY(-50%) scale(5);
}
```

当经过 500 毫秒的延时之后，心形图案就会慢慢消失，这是由 `CSSTransitionGroup` 组件的 `transitionEnterTimeout` 属性值决定的。

关于 React 组件动画的资料，请查看官方文档 [Animation](#)

欢迎添加 Peter 的微信：happypeter1983

冀ICP备15007992号-3