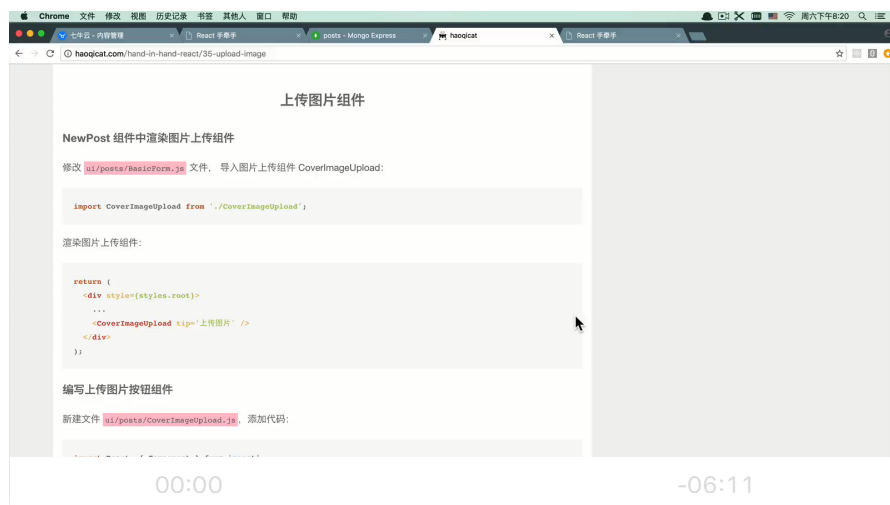


[首页](#) [目录](#)[退出](#)

## 第六章：用户注册功能实现

后端编写用户注册接口

前端编写用户注册表单

## 上传图片组件

### NewPost 组件中渲染图片上传组件

修改 `ui/posts/BasicForm.js` 文件，导入图片上传组件 `CoverImageUpload`：

```
import CoverImageUpload from './CoverImageUp'
```

渲染图片上传组件：

```
return (  
  <div style={styles.root}>  
    ...  
    <CoverImageUpload tip='上传图片' />  
  </div>  
)
```

```
);
```

## 编写上传图片按钮组件

新建文件 `ui/posts/CoverImageUpload.js`，添加代码：

```
import React, { Component } from 'react';
import ContentAdd from 'material-ui/svg-icons/communication/content-add';

class CoverImageUpload extends Component {
  getStyles() {
    return {
      uploadWrapper: {
        marginTop: '20px',
        marginBottom: '30px',
        width: '180px',
        border: '1px solid #ddd',
        height: '180px',
        backgroundColor: '#f8f8f8',
        textAlign: 'center'
      },
      uploadLabel: {
        display: 'block',
        height: '20px',
        lineHeight: '20px',
        fontSize: '13px',
        paddingTop: '80px',
        paddingBottom: '80px',
        cursor: 'pointer'
      },
      svg: {
        width: '20px',
        height: '20px'
      }
    };
  }
}
```

```

    },
    uploadText: {
      display: 'inline-block',
      verticalAlign: 'top'
    },
    uploadButton: {
      display: 'none'
    }
  }
}

render() {
  const styles = this.getStyles();
  return (
    <div style={styles.uploadWrapper}>
      <label style={styles.uploadLabel} htmlFor={this.props.uploadId}>
        <ContentAdd style={styles.svg} />
        <span style={styles.uploadText}>{this.props.uploadText}</span>
      </label>
      <input type='file' id='imageUploadButton' />
    </div>
  );
}
}

export default CoverImageUpload;

```

显示一个好看的上传按钮，不过点击上传按钮之后，虽然能够上传文件，但是页面中看不到任何效果

## 预览图片

初始化一个 `image state`，用来保存上传的图片，代码如下：

```
constructor(props) {  
  super(props);  
  this.state = {  
    image: ''  
  };  
}
```

然后给上传文件输入框添加一个 `onChange` 属性，当输入框的内容改变的时候，会触发 `handleChange` 事件处理函数：

```
return (  
  ...  
  <input type='file' id='imageUploadBtn' onChange={this.handleChange}/>  
);
```

接下来，定义 `handleChange` 事件处理函数：

```
handleChange(event) {  
  const file = event.target.files[0];  
  if (!file.type.match('image.*')) {  
    console.log('请上传图片');  
  } else {  
    const reader = new FileReader();  
    reader.onload = (event) => {  
      this.setState({  
        image: event.target.result,  
      });  
    }  
    reader.readAsDataURL(file);  
  }  
}
```

## FileReader.onload

## FileReader.readAsDataURL

更改最外围 `<div>` 标签的样式，把上传的图片设置为它的背景图：

```
uploadWrapper: {  
  ...  
  backgroundImage: 'url(' + this.state.image  
  backgroundColor: 'cover',  
  backgroundPosition: 'center center',  
  backgroundRepeat: 'no-repeat'  
},
```

更改 `<label>` 标签的样式，让其隐藏：

```
uploadLabel: {  
  display: this.state.image ? 'none' : 'block'  
  ...  
},
```

## 更改上传的图片

添加一个更改上传图片的小图标：

```
return (  
  ...  
  <label style={styles.uploadLabelAdd} htmlFor={id}>  
    <ContentAdd />  
  </label>
```

```
    <input ... />
  );
```

设置这个小图标的样式，其默认是隐藏的，当上传的图片显示之后，小图标会出现杂图片的右上角：

```
uploadLabelAdd: {
  display: this.state.image ? 'block' : 'none',
  backgroundColor: '#ddd',
  height: '24px',
  position: 'absolute',
  top: '0',
  right: '0',
  cursor: 'pointer'
}
```

更改最外围 `<div>` 标签的样式：

```
uploadWrapper: {
  ...
  position: 'relative'
},
```

欢迎添加 Peter 的微信：happypeter1983

冀ICP备15007992号-3