

[首页](#) [目录](#)[退出](#)

第一章：后端搭建 Express API 服务

创建一个简单的 Express 应用

MongoDB 数据库简单操作

删除服务器上的一篇文章

添加删除文章入口

打开 `ui/posts/PostList.js` 文件，然后给首页文章列表中的每篇文章添加一个 删除 链接：

```
<Link to='javascript:;' style={styles.link} />
```

当点击 删除 链接的时候，只执行 `handleClick` 方法，不会发生页面跳转的行为，该方法完成的功能如下：

```
handleClick(value) {  
  this.setState({id: value});  
}
```

```
    this.refs.dialog.handleOpen();  
  }  
}
```

设置 `state` 变量 `id` 值为要删除文章的 `_id` 号，然后让删除文章对话框显示出来。代码

`this.refs.dialog` 代表删除文章对话框组件，默认是隐藏的，触发它的 `handleOpen()` 方法可以让对话框显示出来。下面初始化 `state` 变量 `id` 为空字符：

```
this.state = {  
  ...  
  id: ''  
}
```

接下来，导入删除文章对话框组件 `DeletePost`，

```
import DeletePost from './DeletePost';
```

然后在文章列表下方添加 `DeletePost` 组件：

```
return (  
  <div>  
    ...  
    <DeletePost ref='dialog' />  
  </div>  
);
```

删除文章对话框

新建文件 `ui/posts/DeletePost.js` 文件，添加代

码:

```
import React, { Component } from 'react';
import Radium from 'radium';

class DeletePost extends Component {
  constructor(props) {
    super(props);
    this.state = {
      show: false
    }
  }

  handleClose() {
    this.setState({show: false})
  }

  handleOpen() {
    this.setState({show: true})
  }

  getStyles() {
    return {
      root: {
        position: 'fixed',
        top: '0',
        left: '0',
        right: '0',
        bottom: '0',
        backgroundColor: 'rgba(0,0,0,.6)',
        display: this.state.show === true ? '
        justifyContent: 'center',
        alignItems: 'center',
        zIndex: '999'
      },
    },
  }
}
```

```
dialog: {
  width: '100%',
  maxWidth: '350px',
  height: '100px',
  padding: '16px',
  backgroundColor: '#fff',
  border: '1px solid #ddd',
  borderRadius: '5px',
  position: 'relative',
  color: '#f44336'
},
actions: {
  position: 'absolute',
  bottom: '16px',
  right: '16px'
},
action: {
  display: 'inline-block',
  paddingLeft: '10px',
  paddingRight: '10px',
  color: '#00bcd4',
  fontSize: '.9em',
  ':hover': {
    cursor: 'pointer'
  }
}
};
}
```

```
render() {
  const styles = this.getStyles();
  return (
    <div style={styles.root}>
      <div style={styles.dialog}>
        <p>确定删除吗? </p>
        <div style={styles.actions}>
```

```

        <div style={styles.action} key='{
        <div onClick={this.handleClose.bj
        </div>
        </div>
    </div>
    );
  }
}

export default Radium(DeletePost);

```

对话框的显示和隐藏状态是由 state 变量 show 通过控制组件样式 display 的属性值实现的，默认为隐藏状态

```
display: this.state.show === true ? 'flex' :
```

当点击 取消 按钮的时候，对话框则消失。

删除数据库中的文章

当点击 确定 按钮的时候，真正的删除文章，把下面一行代码：

```
<div style={styles.action} key='0'>确定</div>
```

更改为：

```
<div onClick={this.handleClick.bind(this)} st
```

然后，定义 `handleClick` 事件处理方法：

```
handleClick() {  
  axios.delete(`${Settings.host}/posts/${this.props._id}`)  
    .then((res) => {  
      this.props.removePosts(this.props.id);  
      this.setState({show: false});  
      console.log(res.data.message)  
    })  
    .catch(error => {  
      if (error.response) {  
        console.log(error.response.data.error);  
      } else {  
        console.log(error.message);  
      }  
    });  
}
```

向服务器 API `${Settings.host}/posts/:post_id` 发送 DELETE 请求，删除 `_id` 号为 `this.props.id` 的文章，文章 `_id` 号由 `PostList` 组件传给 `DeletePost` 组件。若文章在服务器 MongoDB 数据库中删除成功，则执行代码：

```
this.props.removePosts(this.props.id)
```

同时也要删除客户端文章列表中 `_id` 号为 `this.props.id` 的文章。

当然还要导入相关的 `axios` 功能模块，以及配置参数。

```
import axios from 'axios';  
import Settings from '../../settings';
```

删除客户端文章列表中的文章

打开 `ui/posts/PostList.js` 文件，给 `DeletePost` 组件传递属性值：

```
<DeletePost id={this.state.id} removePosts={this.removePosts} />
```

接下来，先导入 `filter` 方法：

```
import filter from 'lodash/fp/filter';
```

然后，定义 `filterPosts` 方法：

```
filterPosts(id) {  
  const posts = filter((post) => {  
    return post._id !== id  
  }, this.state.posts);  
  
  this.setState({ posts: posts })  
}
```

通过 `Lodash` 库提供的 `filter` 方法遍历当前 `state` 变量 `posts` 数组中的所有元素，若某个元素的 `_id` 号与所删除文章的 `_id` 号相匹配，则过滤掉，最终会返回一个新的 `posts` 数组。然后把新生成的 `posts` 数组作为新的状态值赋值给 `state` 变量

`posts`，从而导致 `PostList` 组件重新渲染，删除的文章就消失了。

欢迎添加 Peter 的微信：happypeter1983

冀ICP备15007992号-3