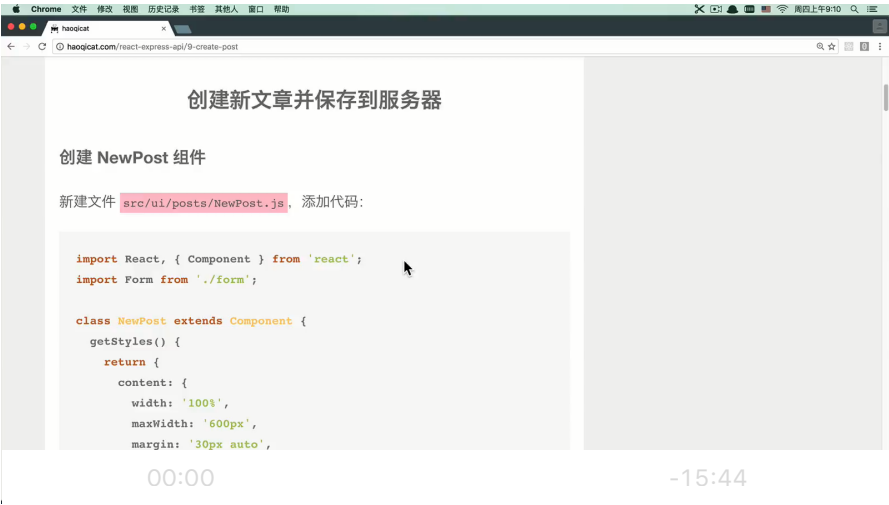


首页 目录

退出



MongoDB 数据库简单操作

Express 对话 MongoDB

Chrome 浏览器中测试 API

构建 REST API

# 创建新文章并保存到服务器

## 创建 NewPost 组件

新建文件 src/ui/posts/NewPost.js , 添加代码:

```
import React, { Component } from 'react';
import Form from './Form';

class NewPost extends Component {
  getStyles() {
    return {
      content: {
        width: '100%',
        maxWidth: '600px',
        margin: '30px auto',
        backgroundColor: '#fff',
        borderRadius: '10px',

```

```
        boxShadow: 'rgba(0, 0, 0, 0.12) 0px 1
    },
    title: {
        fontSize: '1.2em',
        textAlign: 'center',
        paddingTop: '20px'
    }
};

}

render() {
    const styles = this.getStyles();
    return (
        <div style={styles.content}>
            <div style={styles.title}>写文章</div>
            <Form />
        </div>
    );
}

}

export default NewPost;
```

## 构建 NewPost 组件的路由

打开文件 `src/routes.js` , 导入 NewPost 组件:

```
import NewPost from './ui/posts/NewPost';
```

然后构建 NewPost 组件对应的路由, 在

`<IndexRoute ... />` 之后, 添加代码

```
<Route path='/posts/new' component={NewPost}
```

这时到浏览器中，访问地址

`http://localhost:3000/posts/new` 就会看到新添加的 `NewPost` 组件的内容了。

## 添加 Form 组件

Form 组件是用来添加新文章的表单组件，新建文件

`ui/posts/Form.js` 文件，添加代码：

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

class Form extends Component {
  getStyles() {
    return {
      form: {
        padding: '20px 40px',
      },
      div: {
        marginBottom: '10px'
      },
      label: {
        display: 'block',
        fontSize: '.9em',
        color: 'rgba(0,0,0,.6)',
        paddingBottom: '10px'
      },
      input: {
        width: '100%',
        height: '48px',
        border: '1px solid #ddd',
        borderRadius: '5px',
      }
    };
  }
}
```

```
    fontSize: '1em',
    padding: '10px',
    boxSizing: 'border-box',
    ':focus': {
      border: '1px solid #00bcd4',
      outline: 'none'
    }
  },
  actions: {
    textAlign: 'center'
  },
  button: {
    width: '120px',
    height: '36px',
    border: 'none',
    backgroundColor: '#ff4081',
    fontSize: '1em',
    color: '#fff',
    display: 'inline-block',
    margin: '20px auto 0',
    borderRadius: '20px',
    ':hover': {
      cursor: 'pointer'
    },
    ':focus': {
      outline: 'none'
    }
  },
  link: {
    display: 'inline-block',
    marginLeft: '15px',
    fontSize: '1em',
    color: '#00bcd4',
    opacity: '.8',
    textDecoration: 'none'
  }
}
```

```

    };
  }

  render() {
    const styles = this.getStyles();
    return (
      <form style={styles.form}>
        <div style={styles.div}>
          <label style={styles.label}>分类</label>
          <input style={styles.input} key='0' type='text' value={this.state.category}>
        </div>
        <div style={styles.div}>
          <label style={styles.label}>标题</label>
          <input style={styles.input} key='1' type='text' value={this.state.title}>
        </div>
        <div style={styles.div}>
          <label style={styles.label}>内容</label>
          <textarea style={styles.input, {height: 100px}} value={this.state.content}>
        </div>
        <div style={styles.actions}>
          <button type='submit' style={styles.button}>提交
          <Link to='/' style={styles.link}>取消
        </div>
      </form>
    );
  }
}

export default Radium(Form);

```

注意：上述代码中，给每个输入框都添加了一个 **ref** 属性，目的是为了获取输入框中用户填入的数据。另外，借助 **Radium** 工具集让 React 组件的内联样式能够支持 `:focus` 等 CSS 伪类的时候，需要为每个

使用伪类样式的 React 组件添加 `key` 属性，并且每个 `key` 属性值都不一样。

最后，更改 `ui/posts/NewPost.js` 组件，添加代码：

```
import Form from './Form';
```

导入 `Form` 组件，并在 `return` 返回语句中使用 `Form` 组件

```
return (  
  <div style={styles.content}>  
    <div style={styles.title}>写文章</div>  
    <Form label='发布文章' />  
  </div>  
) ;
```

这时到浏览器中，访问地址

`http://localhost:3000/posts/new` 就会看到一个漂亮的 HTML 表单了。

## 获取表单数据

现在实现表单的向 API 服务器提交数据的功能。首先把下面一行代码：

```
<form style={styles.form}>
```

更改为：

```
<form style={styles.form} onSubmit={this.handleSubmit}>
```

当提交表单的时候会触发表单的 `onSubmit` 事件对应的 `handleSubmit` 方法。

接下来，定义 `handleSubmit` 方法：

```
handleSubmit(e) {  
  e.preventDefault();  
  const category = this.refs.category.value;  
  const title = this.refs.title.value;  
  const content = this.refs.content.value;  
  this.props.publishPost({category, title, content});  
}
```

上述代码获取了通过 React 组件的 `ref` 属性值获得了各个输入框的数据之后，组合成一个 JS 对象，然后作为参数传递给 `this.props.publishPost` 方法，最终传送到 API 服务器。这个 `publishPost` 方法是由 `NewPost` 组件传递给 `Form` 组件的，所以我们还需要在 `NewPost` 组件中定义它。

## 创建新文章

打开 `ui/posts/NewPost.js` 文件，首先导入一下功能模块：

```
import axios from 'axios';
```

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

然后，定义 `publishPost` 方法：

```
publishPost(data) {  
  axios.post(`${Settings.host}/posts`, data)  
    .then(res => {  
      console.log(res.data.message);  
      this.context.router.push('/');  
    })  
    .catch(error => {  
      if (error.response) {  
        console.log(error.response.data.error);  
      } else {  
        console.log(error.message);  
      }  
    })  
}
```

向 `${Settings.host}/posts` API 地址发送 Post 请求，提交表单数据。若请求成功，则打印请求返回的信息并跳转到首页。不过让下面一行代码工作：

```
this.context.router.push('/')
```

还得在 `export default NewPost;` 语句之前，添加代码：

```
NewPost.contextTypes = {  
  router: React.PropTypes.object.isRequired  
};
```



至此，创建新文章的功能就实现了。

## 添加写文章页面操作入口

最后，在 PostList 组件中添加一个进入 写文章 页面的操作入口，打开 `ui/posts/PostList.js` 文件，在文章列表上方添加一个链接按钮：

```
<Link to='/posts/new' style={styles.button}>
```

然后，再添加按钮样式代码：

```
getStyles() {  
  return {  
    ...  
  },  
  button: {  
    display: 'block',  
    margin: '30px auto',  
    width: '120px',  
    height: '36px',  
    lineHeight: '36px',  
    textAlign: 'center',  
    backgroundColor: '#ff4081',  
    fontSize: '1em',  
    color: '#fff',  
    textDecoration: 'none',  
    borderRadius: '20px'  
  }  
}
```

欢迎添加 Peter 的微信: happypeter1983

冀ICP备15007992号-3