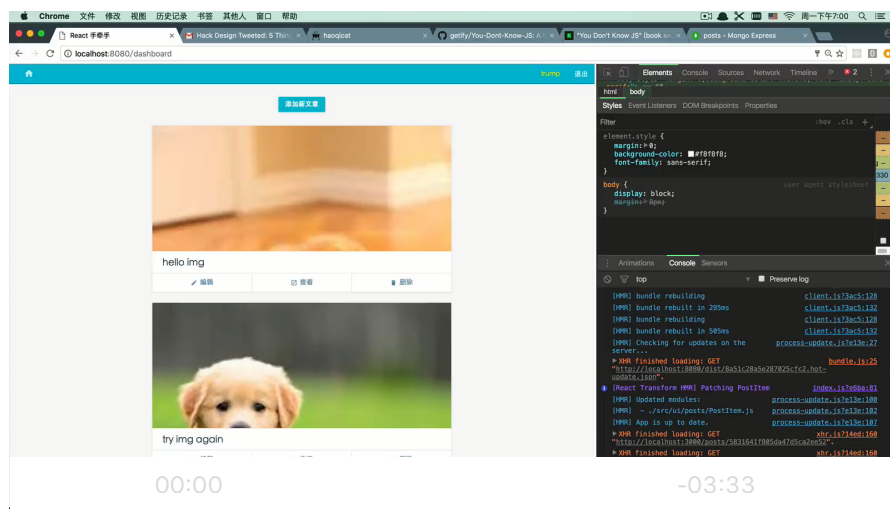


[首页](#) [目录](#)[退出](#)

借助 Redux Thunk 实现异步 Action Creator

React 组件连接
React Store 分发
Redux Action

前端导航栏体现用户
登录状态

实现编辑文章表单组件

前端添加 EditPost 组件路由

打开 `src/routes.js` 文件，导入 EditPost 组件：

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

添加组件对应的路由：

```
<Route path='/posts/:post_id/edit' component=
```

前端添加 EditPost 组件

新建 `ui/posts/EditPost.js` 文件，用来编辑文章，

代码如下:

```
import React, { Component } from 'react';
import RaisedButton from 'material-ui/RaisedButton';
import BasicForm from './BasicForm';
import { connect } from 'react-redux';
import { getPost } from '../../redux/actions';
import isEmpty from 'lodash/fp/isEmpty';

class EditPost extends Component {
  componentWillMount() {
    this.props.getPost(this.props.params.postId);
  }

  handleSubmit(e) {
    // ...
  }

  getStyles() {
    return {
      root: {
        maxWidth: '720px',
        margin: '32px auto 0',
      },
      title: {
        textAlign: 'center',
        color: '#2e4453',
        fontSize: '1.3em'
      },
      submit: {
        textAlign: 'center',
        marginTop: '32px'
      }
    };
  }
}
```

```

render() {
  const styles = this.getStyles();
  return (
    <div style={styles.root}>
      <p style={styles.title}>编辑文章</p>
      <form onSubmit={ this.handleSubmit.bind(
        { !isEmpty(this.props.post) ? <BasicForm
          <div style={styles.submit}>
            <RaisedButton type="submit" label="提交" />
          </div>
        </form>
      </div>
    );
  }
}

EditPost.propTypes = {
  post: React.PropTypes.object.isRequired
}

const mapStateToProps = (state) => ({
  post: state.post
})

export default connect(mapStateToProps, { get

```

编辑文章组件 EditPost 和 添加新文章组件 NewPost 共用 BasicForm 组件。当编辑文章的时候，应该让编辑文章的表单显示已经保存的数据，所以在 EditPost 组件中给 BasicForm 组件传入一个 post 属性，其属性值即为要编辑的文章数据。

修改 BasicForm 组件

给 BasicForm 组件的各个输入框添加一个 `defaultValue` 属性，显示已经保存的文章标题和内容，并把已保存的图片文件名传递给 CoverImageUpload 组件：

```
<div style={styles.root}>
  <TextField ... defaultValue={ this.props.p
  <div style={{marginTop: '15px', marginBott
    <TextField ... defaultValue={ this.props.
  </div>
  <CoverImageUpload ... image={this.props.pos
</div>
```

修改 CoverImageUpload 组件

在 CoverImageUpload 组件中，编写已保存图片在公网上的链接地址，把图片显示出来。导入配置信息：

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

根据从 BasicForm 组件传入的 `image` 属性值，设置 CoverImageUpload 组件内部状态 `image` 的值

```
constructor(props) {
  super(props);
  this.state = {
    image: this.props.image ? `${Settings.hos
  };
}
```

这个 `image` 状态值控制着 `CoverImageUpload` 组件最外层 `<div>` 标签的背景图片是否显示。

添加编辑文章的入口

修改 `PostActionList` 组件，给 `编辑` 标签添加链接，指向编辑文章页面：

```
<Link to={` /posts/${this.props.post._id}/edit`}>
  <EditorModeEdit color='#668eaa' style={style}>
    <span>编辑</span>
  </Link>
```

欢迎添加 Peter 的微信：happypeter1983

冀ICP备15007992号-3