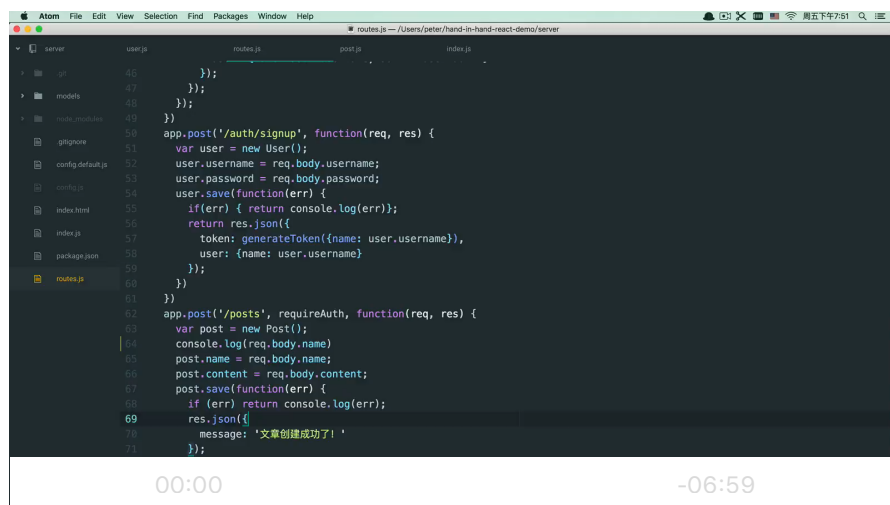


[首页](#) [目录](#)[退出](#)

第一章：搭建后端开发环境

新建一个简单的
Express 应用

Express 应用开发必
备神器 Nodemon

前端展示新添加的文章

修改添加新文章接口的响应数据

上节课中已经可以创建新文章了，不过从服务器端返回的 JSON 数据只有一条提示信息，实际上还需要服务器端把新添加的文章返回给客户端。修改 `server/routes.js` 文件中的添加新文章接口的响应数据：

```
app.post('/posts', requireAuth, function(req, res) {
  ...
  post.save(function(err) {
    ...
    res.json({
      post: post,
```

```
        message: '文章创建成功了!'
      });
    });
  })
```

这样，客户端 Redux store 中的 `posts` 中才会有数据。

DashBoard 组件中展示文件列表

修改 `src/ui/DashBoard.js` 文件，读取 store 中的 `posts` 状态，展示文章列表：

```
import { connect } from 'react-redux';
import PostItem from './posts/PostItem';

class Dashboard extends Component {
  render() {
    ...
    const PostList = this.props.posts.map((post) => {
      return <PostItem key={i} post={post} />
    });

    return (
      <div style={styles.root}>
        ...
        { PostList }
      </div>
    );
  }
}

Dashboard.propTypes = {
  posts: React.PropTypes.array.isRequired
```

```
}

function mapStateToProps(state) {
  return {
    posts: state.posts
  };
}

export default connect(mapStateToProps)(DashI
```

编写 PostItem 组件

新建文件 `ui/posts/PostItem.js`，添加代码：

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

class PostItem extends Component {
  getStyles() {
    return {
      root: {
        marginLeft: 'auto',
        marginRight: 'auto',
        backgroundColor: '#fff',
        border: '1px solid rgba(200, 215, 225, 0.5)',
        boxShadow: '0 0 1px 2px #e9eff3',
        marginBottom: '24px',
        position: 'relative'
      },
      content: {
        padding: '16px 24px 12px',
        lineHeight: '1.3em'
      },
      name: {
```

```
        color: '#2e4453',
        fontWeight: '600',
        fontSize: '1.2em',
        textDecoration: 'none'
      }
    }
  }

  render() {
    const styles = this.getStyles();
    return (
      <div style={styles.root}>
        <div style={styles.content}>
          <div style={styles.name}>
            {this.props.post.name}
          </div>
        </div>
      </div>
    );
  }
}

export default Radium(PostItem);
```

试着添加一篇新文章，这篇文章就会在 /dashboard 页面中显示出来了。

编写 Home 组件

Home 组件是本案例首页要渲染的组件，Home 组件也会展示文章列表。新建文件 ui/Home.js，添加代码：

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

```
import { connect } from 'react-redux';
import PostItem from './posts/PostItem';

class Home extends Component {
  render() {
    const styles = {
      root: {
        maxWidth: '720px',
        margin: '30px auto'
      }
    }

    const PostList = this.props.posts.map((post) => {
      return <PostItem key={post.id} post={post} />
    });

    return (
      <div style={styles.root}>
        { PostList }
      </div>
    );
  }
}

Home.propTypes = {
  posts: React.PropTypes.array.isRequired
}

function mapStateToProps(state) {
  return {
    posts: state.posts
  };
}

export default connect(mapStateToProps)(Home)
```

添加 Home 组件对应的路由

修改路由文件 `src/routes.js`，导入 `IndexRoute` 组件把 Home 组件设置为首页渲染的组件：

```
import { ..., IndexRoute } from 'react-router'
```

然后，修改路由模块，设置 Home 组件对应的路径

/

```
<Route path="/" component={App}>
  <IndexRoute component={Home} />
  ...
</Route>
```

本节课程结束之后，存储在 `store` 中的 `posts` 数据就展示在页面中了，不过有一个问题，刷新页面之后，存储在 `store` 中的数据会被清除，页面中的文章列表也不复存在。下一节课程，我们将解决这个问题。

欢迎添加 Peter 的微信：happypeter1983

冀ICP备15007992号-3