



添加路由

美丽的心

第三章 Redux 来了

Redux 核心概念之 Store

Redux 核心概念之 Actions

Redux 核心概念之 Reducers

本节视频会揭开 Redux **Reducers** 的神秘面目。

一个 reducer 就是一个 JS **纯函数**，这个函数有两个参数，一个是上一个 state 值，另一个就是上节视频介绍的 Redux **action**，它的返回值 state。一般一个 state 变量对应着

一个 reducer 文件，本案例有 courses 和 comments 两个 state，下面我们先定义 courses reducer。

编写 courses reducer

新建一个文件 reducers/courses.js，添加代码：

```
function courses(state = [], action) {  
  console.log(state, action);  
  return state;  
}  
  
export default courses;
```

上面代码就定义了一个 courses reducer，不过这个 reducer 并没有对上一个 state 做任何处理，只是打印 state 和 action 的值，并返回 state 值

编写 comments reducer

接下来新建一个文件 reducers/comments.js，添加代码：

```
function comments(state = [], action) {  
  console.log(state, action);  
  return state;  
}  
  
export default comments;
```

合并 courses 和 comments 两个 reducers

新建一个文件 reducers/index.js，添加代码：

```
import { combineReducers } from 'redux';  
import { routerReducer } from 'react-router-redux';  
  
import courses from './courses';  
import comments from './comments';
```

```
const rootReducer = combineReducers({courses, comments, routing: router

export default rootReducer;
```

因为在 Redux 项目中只有唯一的一个状态树（state tree），所以通过 `combineReducers` 把刚才定义的 `courses reducer` 和 `comments reducer` 合并起来，生成一个 `rootReducer` 提供给创建 Redux store 的接口 `createStore` 使用。

`routerReducer` 作用是把路由状态添加到状态树中（用于开发调试，可以不添加），其对应的 state 变量名是 `routing`，最终本案例的状态树是这样的：

```
{
  courses: courses,
  comments: comments,
  routing: routerReducer
}
```

这个 JS 对象的每一个 key 对应着存储在 store 中的一个状态变量的名字，每一个 value 则对应着一个 reducer，决定着状态变量值。

欢迎添加 Peter 的微信：happypeter1983

冀ICP备15007992号-3