

Chrome 文件 修改 视图 历史记录 书签 其他人 窗口 帮助

haoqicat.com/redux-tower/6-actions

Redux 核心概念之 Actions

我们已经创建了一个 Redux store 容器，也知道了 store 容器用来储存应用中的 state，那怎么来更新这些 state，就需要先了解一些关于 Redux actions 的事情。下面就开始说说与 Redux actions 相关的那些事。

什么是 Actions

Actions 描述了应用中所发生的事情，actions 可不是唯一的，因为应用中会发生很多事情，比如说本案例中会发生的事件就有点赞课程、添加评论等等。那怎么用代码来代表所发生的事情呢？一个 action 就是一个平常的 JavaScript 对象，可以这样定义：

```

{
  type: 'INCREMENT',
  number: 1
}

```

一个 action 对象必须要有一个 type 属性，用来表明所发生事情的类型，type 属性值一般是大的字符串常量。除 type 属性之外，还可以带有其它的属性，如代码中的 index，代表事情发生时产生的数据信息。

什么是 Action Creators

Redux 核心概念之 Store

Redux 核心概念之 Actions

Redux 核心概念之 Reducers

第四章 React 与 Redux 相识相知

React 和 Redux 搭建友谊的小船

React Devtools 中演示 Redux 数据流

读取 Store 中的 Courses State

更新 Store 中的 Courses State

第五章 实现课程评论功能

显示课程评论

添加和删除评论

第六章 实用的开发者工具

热加载 Redux Reducers

0:00 / 4:29

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 Actions

我们已经创建了一个 Redux store 容器，也知道了 store 容器用来储存应用中的 state，那怎么来更新这些 state，就需要先了解一些关于 Redux actions 的事情。下面就开始说说与 Redux actions 相关的那些事。

什么是 Actions

Actions 描述了应用中所发生的事情，`actions` 可不是唯一的，因为应用中会发生很多事情，比如说本案例中会发生的事件就有点赞课程、添加评论等等。那怎么用代码来代表所发生的事情呢？一个 `action` 就是一个平常的 JavaScript 对象，可以这样定义：

```
{  
  type: 'INCREMENT',  
  number: 1  
}
```

一个 `action` 对象必须要有一个 `type` 属性，用来表明所发生事情的类型，`type` 属性值一般是大写的字符串常量。除 `type` 属性之外，还可以带有其它的属性，如代码中的 `index`，代表事情发生时产生的数据信息。

什么是 Action Creators

Action creators 就是用来创建 `actions` 的函数，函数返回值是 `action` 对象类型。一个 `action creator` 的定义是类似这样的：

```
function increment(number) {  
  return {  
    type: 'INCREMENT',  
    number  
  }  
}
```

接下来，我们就开始编写本案例需要的 `action creators`。新建一个文件 `actions/index.js`，添加如下代码：

```
// action types  
  
export const INCREMENT_LIKES = 'INCREMENT_LIKES';  
export const ADD_COMMENT = 'ADD_COMMENT';  
export const REMOVE_COMMENT = 'REMOVE_COMMENT';
```

首先定义本案例所需要的三个 `action` 类型，当项目中有很多 `action` 类型的时候，可以单独创建一个文件来存储所有的 `action` 类型。

然后，定义点赞课程的 `action creator`：

```
export function increment(index) {  
  return { type: INCREMENT_LIKES, index }  
}
```

其中，`index` 代表被点赞课程在课程数组中的索引号

接下来，定义添加评论的 `action creator`：

```
export function addComment(courseId, author, comment) {  
  return { type: ADD_COMMENT, courseId, author, comment }  
}
```

其中，`courseId` 是被评论的课程 `id` 号，`author` 是评论者的名字，`comment` 是评论的内容

最后，定义删除评论的 `action creator`：

```
export function removeComment(courseId, i) {  
  return { type: REMOVE_COMMENT, i, courseId }  
}
```

其中，`courseId` 是被删除评论隶属课程的 `id` 号，`i` 是指被删除评论在隶属课程的评论数组中的索引号

至此，我们已经知道了 `Redux actions` 的庐山真面目，怎么在项目中使用它们呢？这就涉及到 `Redux reducers` 的相关知识了，`reducers` 会利用 `actions` 来更新 `store` 容器中存储的 `state`。

欢迎添加 Peter 的微信: happypeter1983

冀ICP备15007992号-3