链滴

redux 概念简述

作者：limanting

原文链接：https://ld246.com/article/1629039824289

来源网站：链滴

如果已经简单了解过redux的概念，可以直接看redux的快速使用。

redux的核心概念

掌握了 redux的核心概念，才能更加快速的领悟 redux的写法。

还记得react的哲学吗？ 将设计好的 UI 划分为组件层级、创建静态版本、确定UI state 的最小（且整）表示、添加反向数据流。

https://react.docschina.org/docs/thinking-in-react.html

redux就是将除了静态版本外的所有 state（状态）进行管理。所以，在使用 redux的过程中，这些与tate（状态）有关的内容将全部在redux中处理。

redux三大原则及其在代码中的体现

单一数据源

整个应用的 state 被储存在一棵 三大原则object tree 中，并且这个 object tree 只存在于唯一一个 store 中。

所以项目中应该有一个单独的 store文件夹，放置所有的state及其相关内容。如果将整个应用的所有state都放在一起，难免使得可读性变差，很难区分某个store究竟是为哪个页面部分服务的。

为了解决上述问题，我们可以将需要管理状态的页面或者组件的文件夹中加入 store文件夹，将actio、reducer等写在组件或页面中的store中，最后在整个项目的 store中将他们集中起来。

State 是只读的

唯一改变 state 的方法就是触发 action，action 是一个用于描述已发生事件的普通对象。

确保了视图和网络请求都不能直接修改 state，所有的修改都被集中化处理，且严格按照一个接一个顺序执行，因此不用担心 race condition 的出现， Action 就是普通对象而已，因此它们可以被日志印、序列化、储存、后期调试或测试时回放出来。

使用纯函数来执行修改

为了描述 action 如何改变 state tree ，需要编写 reducers

Reducer 只是一些纯函数，它接收先前的 state 和 action，并返回新的 state。刚开始你可以只有一个 reducer，随着应用变大，你可以把它拆成多个小的 reducers，分别独立地操作 state tree 的不同部，因为 reducer 只是函数，你可以控制它们被调用的顺序，传入附加数据，甚至编写可复用的 reduce 来处理一些通用任务，如分页器。

action、reducer 、store的简单介绍

Action

Action 是把数据从应用传到 store 的有效载荷，是 store 数据的唯一来源。一般来说你会通过 store.dspatch() 将 action 传到 store。

Action 创建函数 就是生成 action 的方法。

Reducer

Reducers 指定了应用状态的变化如何响应 actions 并发送到 store 的，记住 actions 只是描述了有情发生了这一事实，并没有描述应用如何更新 state。

Action 处理

reducer 就是一个纯函数，接收旧的 state 和 action，返回新的 state

永远不要在 reducer 里做这些操作：

1. 修改传入参数
2. 执行有副作用的操作，如 API 请求和路由跳转
3. 调用非纯函数，如 Date.now() 或 Math.random()

只要传入参数相同，返回计算得到的下一个 state 就一定相同。没有特殊情况、没有副作用，没有 API 请求、没有变量修改，单纯执行计算。

Store

store 是把 action 和使用 reducers 来根据 action 更新 state 的用法 联系到一起的对象

store 的作用

1. 维持应用的 state
2. 提供 getState() 方法获取 state
3. 提供 dispatch(action) 方法更新 state
4. 通过 subscribe(listener) 注册监听器
5. 通过 subscribe(listener) 返回的函数注销监听器