

# 购物车组件

redux的用法

购物车，全选反选、多选etc

从零开始手写一个redux



by eric wen

# 分析UI

- 1、在商品详情页，添加加入购物车逻辑
- 2、可单选，多选，全选，反选
- 3、监听数量组件，计算总数
- 4、从redux读购物车数据

# 组件设计

|        |  |  |
|--------|--|--|
| props  |  |  |
| Events |  |  |

# 组件开发

## 组件开发

### 1、引入redux，定义store reducer

- npm install @reduxjs/toolkit react-redux
- 在store文件夹新建index.js，引用toolkit的configureStore，初始化
- 在index.js我们应用程序入口，将redux当成父组件
- 初始化购物车store，并创建对应着的reducer，toolkit会帮我们返回action，这个action到时直接使用就行
- 最后把刚创建的store给到store实例

### 2、在商品详情页添加逻辑，点击加入购物车，即将redux 存入一个商品

### 3、从redux获取数据，并渲染

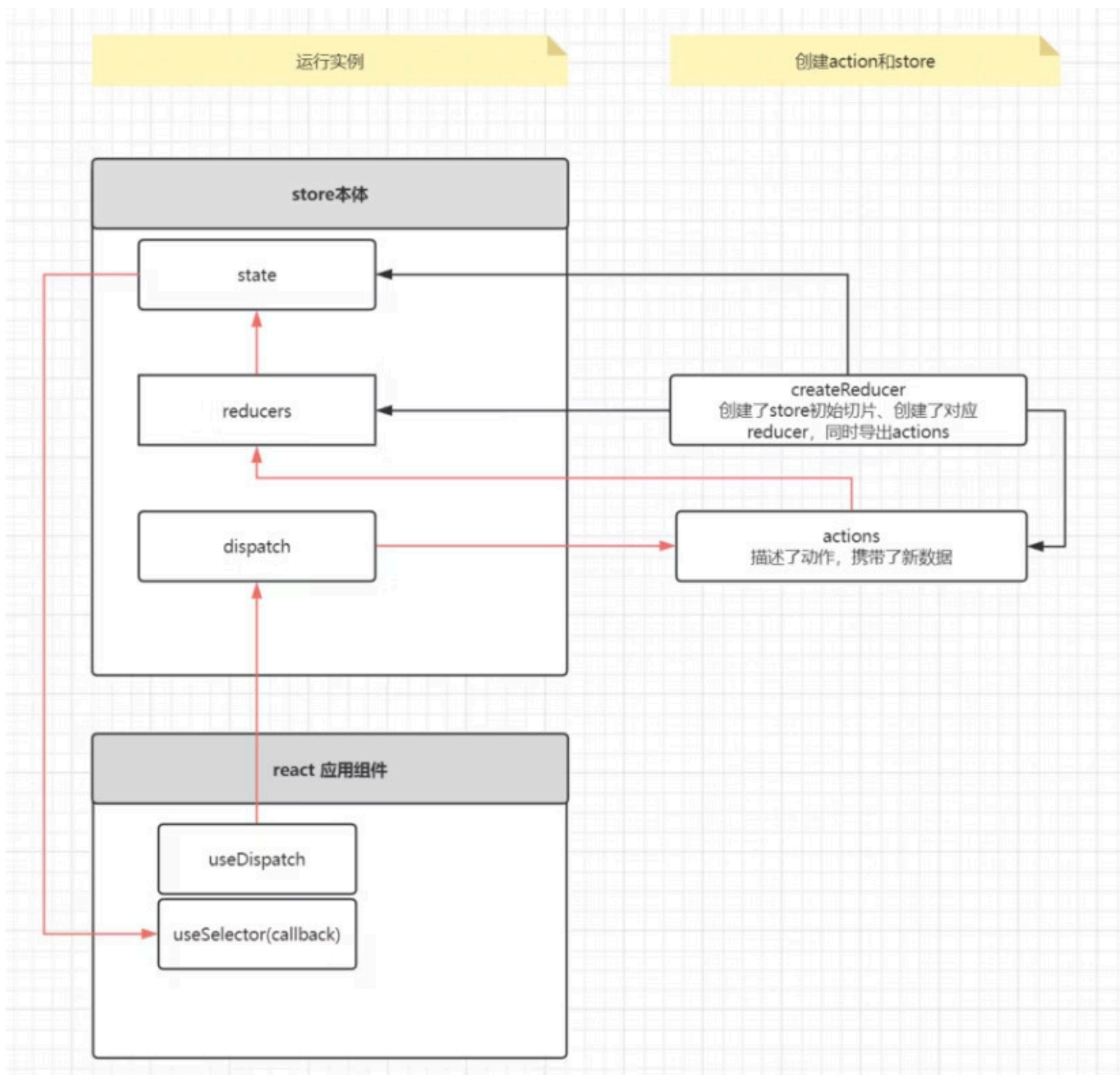
### 4、反选、全选联动逻辑

# redux基础知识

1、基本概念

2、用法

# 手写redux



## 功能实现

- 1、store本质上是观察者订阅机制，通过useSelector订阅，实现useSelector，即订阅分发逻辑
- 2、实现useDispatch，dispatch接受一个action，action是一个对象，有type和payload两个数据，store实例，即redux调用reducer执行，处理action的payload，修改state。
- 3、state改变，分发给订阅方新的值，触发调用方重渲染，注意，state必须引用原生useState
- 4、CreateReducer函数，本质上是生成reducer纯函数丢给store实例，让store能在dispatch的时候调用，用来处理action。

Reducer和action是一对多的绑定关系，要不然store不知道如何处理action携带的数据。所以在这个函数里面，使用方要编写action的逻辑。

此函数会自动生成reducer，并传递给store。

如此，createReducer实际上创建的是store切片，以及对应切片的reducer和action，独立。

用户可以用此函数创建多个store切片，以方便state的数据分割和维护。

# 替换我们的redux