Redux

1. Why Redux?

Redux是一个web应用的状态管理容器

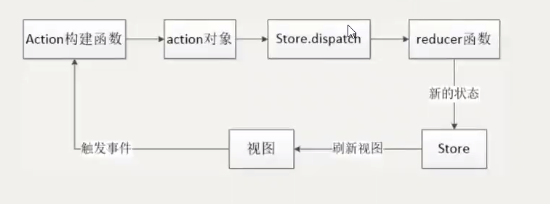
·React本身只是一个试图框架

·随着Js单页面应用开发日趋复杂，js需要管理比任何时候都要多的state

·让web应用的行为可以预测——Redux目的·

【视图模板：通过模板把视图动态化，能够根据它不同传入信息生成不同html页面】

1. Redux新概念
2. Redux数据流：严格的单向数据流是Redux架构的设计核心



说明： 通过store来管理state

Reducer决定了状态如何改变

·单向数据流：试图上的数据不能流向JavaScript里面的数据只能是通过state里面的数据来影响它的input

·双向数据绑定：视图数据和js的数据能相互影响

·单向数据绑定：你只能通过单方面影响另一方面

1. Store

1）概念：Store用来维持应用所有的state树的一个对象。改变store内state的唯一途径是对它的dispatch一个action

应用中应有且只有一个Store

1. 获取：通过createStore(reducer)接口，用于创建store对象
2. 方法：i) getState()：获得state

Ii)dispatch(action)：触发一个action

Iii)subscribe(listener)：用于添加监听函数

1. Action

Action唯一可以改变state的方式，Action本质上是JavaScript普通对象。我们约定，action内必须使用一个字符串类型的type字段来表示将要执行的动作

1. Reducer

Action只是描述了有事情发生这一事实，并没有指明应用如何更新state，而这正是reducer需要做的事情。reducer本质是一个纯函数，用于描述action如何改变state

纯函数： 1、不对外界产生任何影响

1. 不对输入结果产生影响

Reducer函数的纯度决定执行效率

1. Git

Git是代码管理工具

1、分布式代码管理工具：么一部分代码都是核心，不需要联网也能使用，提供merge工具

2、代码仓库： 本地仓库——建立：在本地执行git init

远端仓库