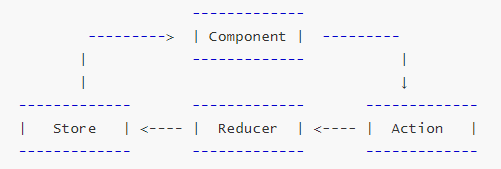
**一、简答题**

1、在学习了Redux之后，请根据所给图片信息，进行相关流程描述  
（提示：以Action作为起始点）  


|  |
| --- |
| **考察：Redux的实现流程**  答：用户页面行为触发一个Action，然后，Store 自动调用 Reducer，并且传入两个参数：当前 State 和收到的 Action。Reducer 会返回新的 State 。每当state更新之后，view视图（Component）会根据state触发重新渲染。（合理即可） |

2、Redux API 和 Store API你都使用过哪些，并简答说明下API的作用

|  |
| --- |
| **考察：Redux和Store中的API**  答：Redux API中使用过createStore和applyMiddleware   1. createStore: 创建一个 Redux [store](https://www.redux.org.cn/docs/api/Store.html) 来以存放应用中所有的 state 2. applyMiddleware: 通过middleware扩展Redux的一个方式   Store API中使用过getState、dispatch和subscribe ① getState: 返回应用当前的 state 树   1. dispatch: 触发 state 变化的唯一途径 2. Subscribe: 添加一个变化监听器，每当 dispatch action 的时候就会执行 |

3、在学习了Redux之后我们又学习了React-Redux，React-Redux帮助我们解决优化了哪些地方？再说说你对React-Redux的两个核心API的理解

|  |
| --- |
| **考察：React-Redux的优势及React-Redux的2个核心API（Provider和Connect）**  答：使用React-Redux解决了每次都需要初始化state手动监听数据的步骤，优化了数据的层层传递问题。React-Redux的两个核心API分别是： ① Provider: 从最外部封装整个应用，并向connect模块传递store   1. Connect: 1.包装原组件，将state和action通过props的方式传入到原组件内 2.监听store tree变化，使其包装的原组件可以响应state变化 |

4、你知道中间件redux-thunk吗，说说它的应用场景及对你的帮助吧

|  |
| --- |
| **考察：redux-thunk的应用场景及作用**  答：redux-thunk应用场景: Redux store仅支持同步数据流，通过使用thunk等中间件可以帮助在Redux中实现异步性，异步地生成Ajax请求以获取action  redux-thunk作用: 帮助我们统一了异步和同步 action 的调用方式，把异步过程放在 action 级别解决，对 component 调用没有影响 |

5、学习vue时我们学习了vuex，学习react时我们学习了redux，他们都是一个状态管理系统。在vuex和redux的使用体验上你更喜欢使用哪一个，请思考后，详细说明下你更喜欢ta的理由

|  |
| --- |
| **考察：vuex和redux的区别及主观体验**  答：（合理即可）  相同点：  1.数据驱动视图，提供响应式的视图组件  2.都有virtual DOM， 组件化开发，通过props参数进行父子组件数据的传递，都实现webComponents规范  3.都支持服务端渲染  4.都有native解决方案，reactnative（facebook团队） vs weex（阿里团队）  不同点：  1.vuex是一个针对VUE优化的状态管理系统，而redux仅是一个常规的状态管理系统（Redux）与React框架的结合版本。  2.开发模式：React本身，是严格的view层，MVC模式；Vue则是MVVM模式的一种方式实现  3.数据绑定：Vue借鉴了angular，采取双向数据绑定的方式；React，则采取单向数据流的方式  4.数据更新：Vue采取依赖追踪，默认是优化状态：按需更新；  React在则有两种选择：  1）手动添加shouldComponentUpdate，来避免冗余的vdom，re-render的情况  2）Components 尽可能都用 pureRenderMixin，然后采用 redux 结构 + Immutable.js  5.社区：react相比来讲还是要大于vue，毕竟背后支撑团队不同。  facebook vs 个人！当然目前vue的增长速度是高于react的增速，不知道未来的发展趋势是如何。  总之：期待构建一个大型应用程序——选择React，期待应用尽可能的小和快——选择Vue |

**二、编程题**

**1、**已知三个组件App、AppSon、AppGrSon，存在嵌套关系，根据已给出的模板，用上所学的redux和react-redux，实现以下功能：

**功能一：**AppSon和AppGrSon中的有初始数据并实时更新

**功能二：**在App中的表单输入内容后，然后点击Send发送

**功能三：**在AppSon中记录所有发送的信息记录  
记录格式为：{时间}——{信息}——{状态}

**功能四：**在AppGrSon中接收App发送过来的信息，信息均为未读，当点击已读按钮时，信息呈现已读 并更新信息时间

★**参考代码如下：**



**2、**根据已给模板及服务端代码，使用所学的redux-thunk，完成一个简单的请求数据并渲染功能。

**请求接口地址：**http://localhost:8080/getNum

**要求如下：**

1. 点击获取随机码按钮
2. 获取到数据之前，视图变浅
3. 获取到数据之后，视图恢复并渲染

**★参考代码如下：**



(服务端代码)  
