React和Redux的连接react-redux

react-redux提供两个关键模块：Provider和connect。

react+react-router+webpack+redux+es6+ant-design

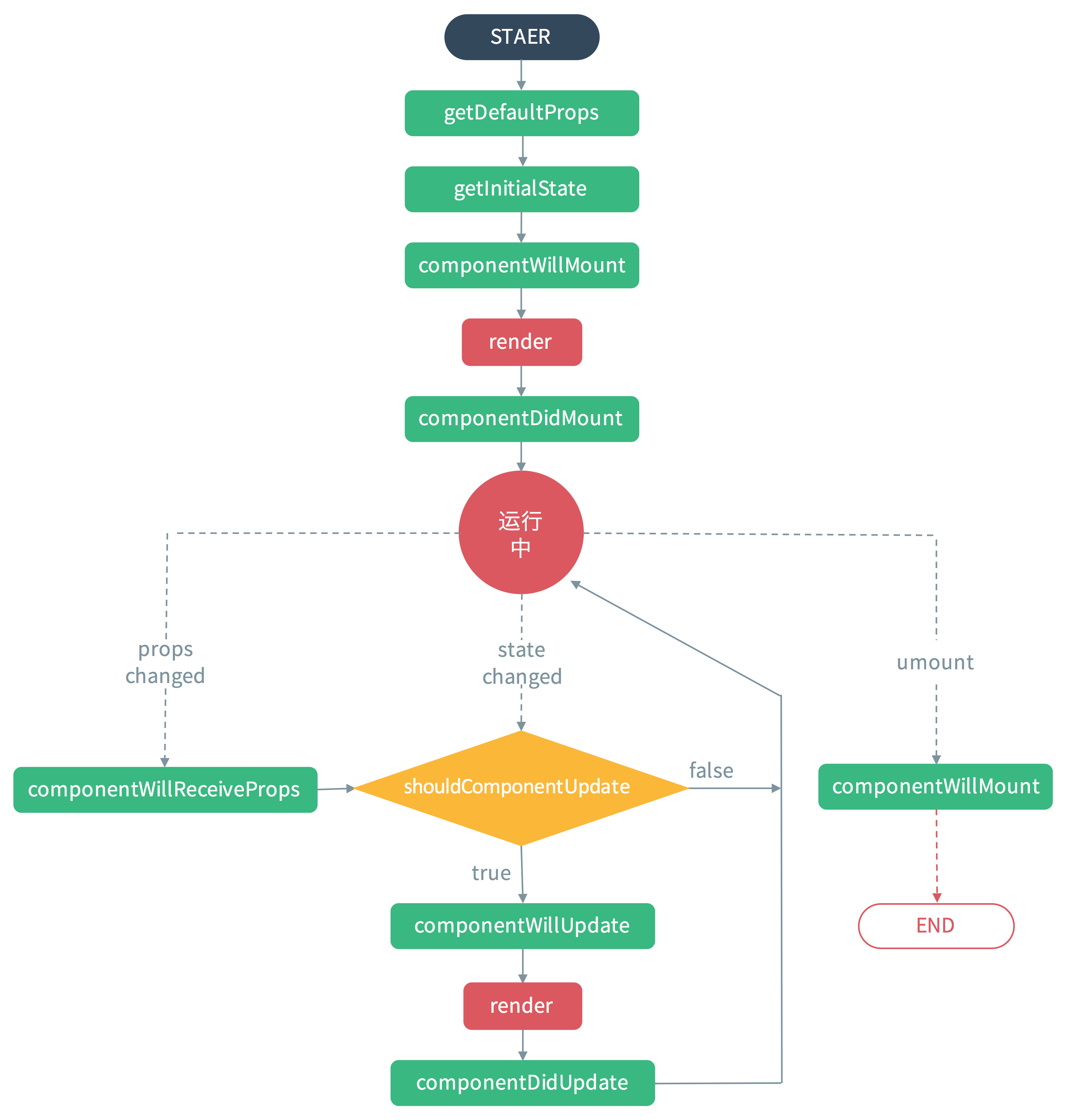
react生命周期

getDefaultProps，getInitialState， componentWillMount，

render，componentDidMount，componentWillReceiveProps，

shouldComponentUpdate，componentWillUpdate，componentDidUpdate，

componentWillUnmount，



**组件在初始化时会触发5个钩子函数：**

**1、getDefaultProps()**

设置默认的props，也可以用dufaultProps设置组件的默认属性。

**2、getInitialState()**

在使用es6的class语法时是没有这个钩子函数的，可以直接在constructor中定义this.state。此时可以访问this.props。

**3、componentWillMount()**

组件初始化时只调用，以后组件更新不调用，整个生命周期只调用一次，此时可以修改state。

**4、 render()**

react最重要的步骤，创建虚拟dom，进行diff算法，更新dom树都在此进行。此时就不能更改state了。

**5、componentDidMount()**

组件渲染之后调用，可以通过this.getDOMNode()获取和操作dom节点，只调用一次。

**在更新时也会触发5个钩子函数：**

**6、componentWillReceivePorps(nextProps)**

组件初始化时不调用，组件接受新的props时调用。

**7、shouldComponentUpdate(nextProps, nextState)**

react性能优化非常重要的一环。组件接受新的state或者props时调用，我们可以设置在此对比前后两个props和state是否相同，如果相同则返回false阻止更新，因为相同的属性状态一定会生成相同的dom树，这样就不需要创造新的dom树和旧的dom树进行diff算法对比，节省大量性能，尤其是在dom结构复杂的时候。不过调用this.forceUpdate会跳过此步骤。

**8、componentWillUpdate(nextProps, nextState)**

组件初始化时不调用，只有在组件将要更新时才调用，此时可以修改state

**9、render()**

不多说

**10、componentDidUpdate()**

组件初始化时不调用，组件更新完成后调用，此时可以获取dom节点。

还有一个卸载钩子函数

**11、componentWillUnmount()**

组件将要卸载时调用，一些事件监听和定时器需要在此时清除。

组件化、模块化、开发效率、运行效率、可维护性…

通过react-redux这个库，可以方便的将react和redux结合起来：React负责页面展现，Redux负责维护/更新数据状态。大致过程为：当用户在View中触发事件产生Action，Action 进到 Reducer，Reducer根据Action Type去匹配对应处理的动作，然后返回一个新的状态。View则因为检测到状态更新而进行重绘。Redux只有一个Store负责存放整个App的状态，而唯一能改变状态的方法只有发送Action。

RN、Weex

React:

Ant Design: 支付宝

Material Design ：谷歌



CommonJS

浏览器不兼容CommonJS的根本原因，在于缺少四个Node.js环境的变量

