

redux

redux

- 课堂目标
- 知识要点
- 资源
- 你可能不需要redux
- redux
 - 安装redux
 - redux上手
 - 检查点

课堂目标

1. 理解我们什么时候才会需要redux
2. 掌握redux基本使用

知识要点

2. redux如何使用

资源

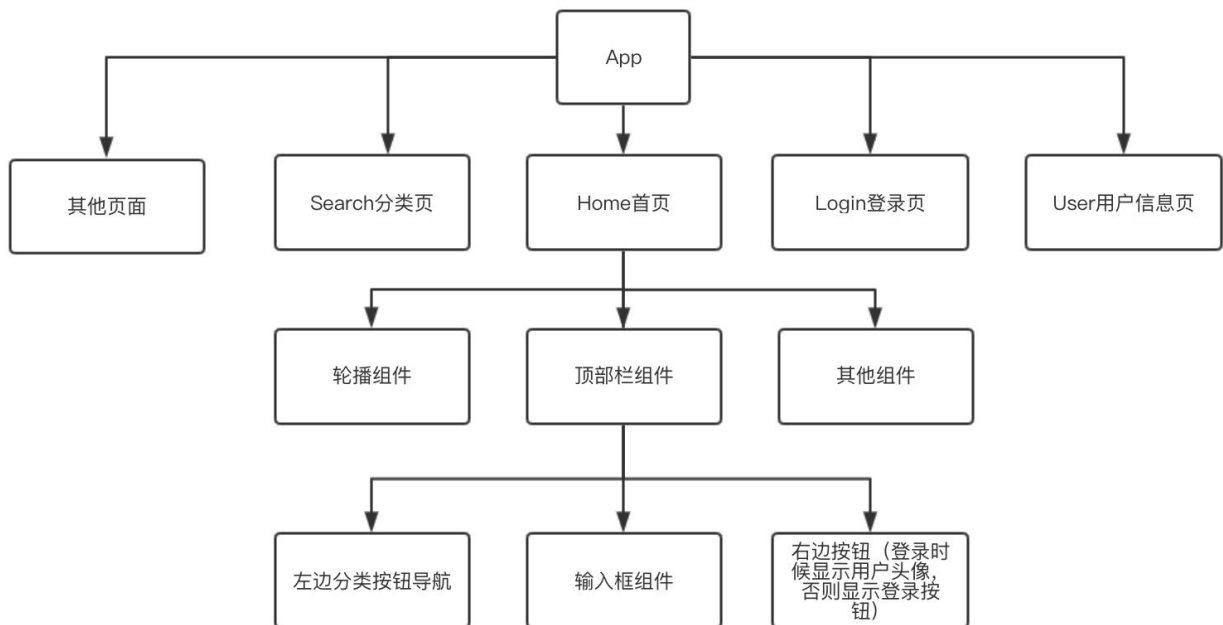
1. [redux](#)

你可能不需要redux

Redux 是负责组织 state 的工具，但你也要考虑它是否适合你的情况。不要因为有人告诉你要用 Redux 就去用，花点时间好好想想使用了 Redux 会带来的好处或坏处。

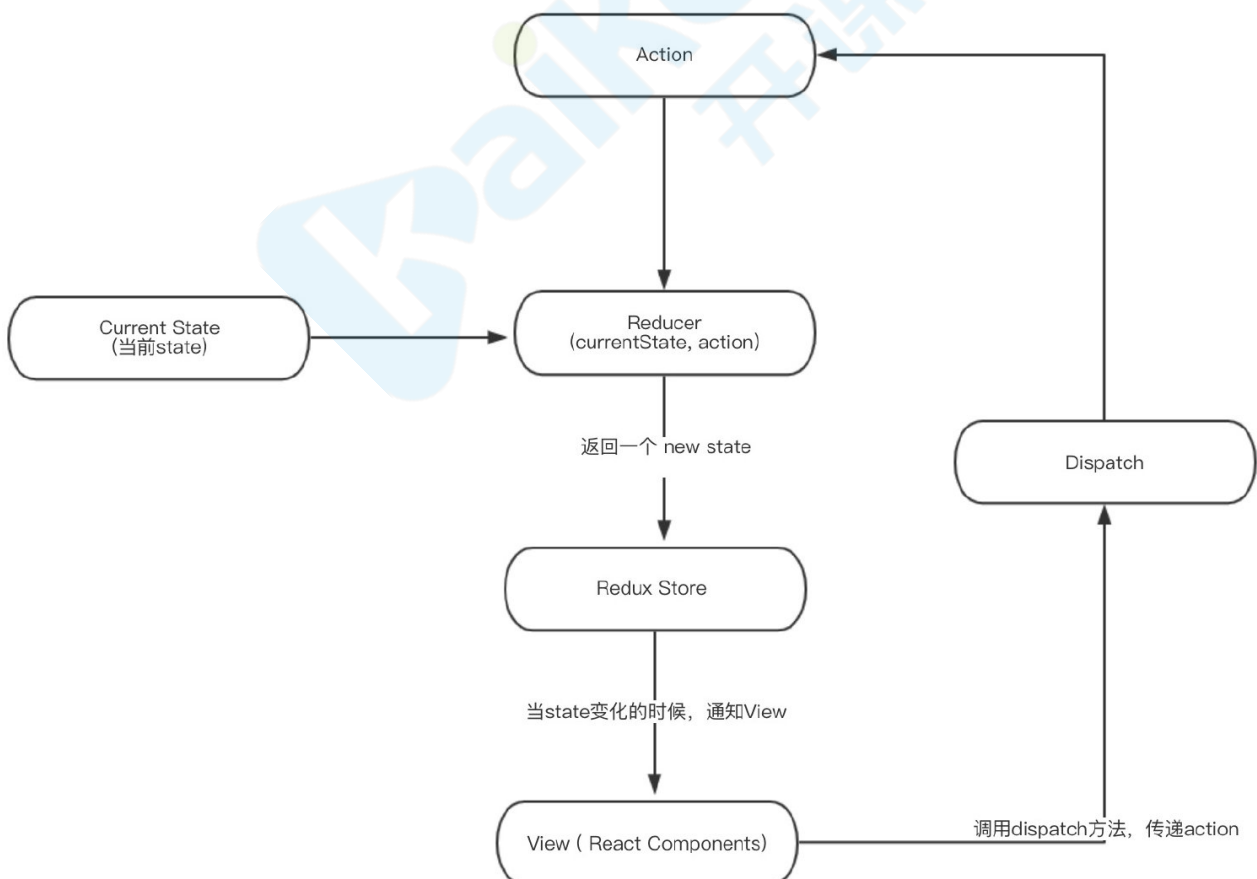
在下面的场景中，引入 Redux 是比较明智的：

- 你有着相当大量的、随时间变化的数据；
- 你的 state 需要有一个单一可靠数据来源；
- 你觉得把所有 state 放在最顶层组件中已经无法满足需要了。
- 某个组件的状态需要共享。



redux

redux 是 JavaScript应用的状态容器，提供可预测化的状态管理。它保证程序行为一致性且易于测试。



安装redux

```
npm install redux --save
```

redux上手

用一个累加器举例

1. 需要一个store来存储数据
2. store里的reducer初始化state并定义state修改规则
3. 通过dispatch一个action来提交对数据的修改
4. action提交到reducer函数里, 根据传入的action的type, 返回新的state

创建store, src/store/ReduxStore.js

```
import {createStore} from 'redux'

const counterReducer = (state = 0, action) => {
  switch (action.type) {
    case 'ADD':
      return state + 1
    case 'MINUS':
      return state - 1
    default:
      return state
  }
}

const store = createStore(counterReducer)

export default store
```

创建ReduxPage

```
import React, { Component } from "react";
import store from "../store/ReduxStore";

export default class ReduxPage extends Component {
  componentDidMount() {
    store.subscribe(() => {
      console.log("subscribe");
      this.forceUpdate();
      //this.setState({});
    });
  }
  add = () => {
    store.dispatch({ type: "ADD" });
  };
  minus = () => {
```

```

    store.dispatch({ type: "MINUS" });
  };
  render() {
    console.log("store", store);
    return (
      <div>
        <h3>ReduxPage</h3>
        <p>{store.getState()}</p>
        <button onClick={this.add}>add</button>
        <button onClick={this.minus}>minus</button>
      </div>
    );
  }
}

```

如果点击按钮不能更新，因为没有订阅(subscribe)状态变更

还可以在src/index.js的render里订阅状态变更

```

import store from './store/ReduxStore'
const render = ()=>{

  ReactDOM.render(
    <App/>,
    document.querySelector('#root')
  )
}
render()

store.subscribe(render)

```

检查点

1. createStore 创建store
2. reducer 初始化、修改状态函数
3. getState 获取状态值
4. dispatch 提交更新
5. subscribe 变更订阅