**State**

state表示Web应用的状态，用于保存Web应用的数据，todo 应用（示例应用）的 state 可能长这样：

{

todos: [

{

text: 'Eat food',

completed: true

}, {

text: 'Exercise',

completed: false

}

],

visibilityFilter: 'SHOW\_COMPLETED'}

要想更新 state 中的数据，你需要发起一个 action。

**Action**

Action 就是一个普通 JavaScript 对象用来描述发生了什么。下面是一些 action 的示例：

{ type: 'ADD\_TODO', text: 'Go to swimming pool' }

{ type: 'TOGGLE\_TODO', index: 1 }

{ type: 'SET\_VISIBILITY\_FILTER', filter: 'SHOW\_ALL' }

**Reducer**

action 如何更新 state？这个功能有 reducer 完成。

reducer 是一个接收 state 和 action，并返回新的 state 的函数。如下2个reducer示例：

function visibilityFilter(state = 'SHOW\_ALL', action) {

if (action.type === 'SET\_VISIBILITY\_FILTER') {

return action.filter;

} else {

return state;

}

}

function todos(state = [], action) {

switch (action.type) {

case 'ADD\_TODO':

return state.concat([{ text: action.text, completed: false }]);

case 'TOGGLE\_TODO':

return state.map((todo, index) =>

action.index === index ?

{ text: todo.text, completed: !todo.completed } :

todo

)

default:

return state;

}

}

再开发一个 reducer 调用这两个 reducer，进而来管理整个应用的 state：

function todoApp(state = {}, action) {

return {

todos: todos(state.todos, action),

visibilityFilter: visibilityFilter(state.visibilityFilter, action)

};

}

这差不多就是 Redux 思想的全部。