Store 有以下职责：

* 维持应用的 state；
* 提供 [getState()](https://www.redux.org.cn/docs/api/Store.html" \l "getState) 方法获取 state；
* 提供 [dispatch(action)](https://www.redux.org.cn/docs/api/Store.html" \l "dispatch) 方法更新 state；
* 通过 [subscribe(listener)](https://www.redux.org.cn/docs/api/Store.html" \l "subscribe) 注册监听器;
* 通过 [subscribe(listener)](https://www.redux.org.cn/docs/api/Store.html" \l "subscribe) 返回的函数注销监听器。

## **根据Reducer创建store**

* 根据已有的 reducer 来创建 store 是非常容易的。

import { createStore } from 'redux'

import todoApp from './reducers'

let store = createStore(todoApp)

[createStore()](https://www.redux.org.cn/docs/api/createStore.html) 的第二个参数是可选的, 用于设置 state 初始状态。

let store = createStore(todoApp, window.STATE\_FROM\_SERVER)

## **发起 Actions**

向store发送action，action会被传给reducer，reducer处理action并返回新的state

import {

addTodo,

toggleTodo,

setVisibilityFilter,

VisibilityFilters} from './actions'

// 打印初始状态

console.log(store.getState())

// 注册监听器，每次 state 更新时，监听器都会被调用

// 注意 subscribe() 返回一个函数用来注销监听器

const unsubscribe = store.subscribe(() =>

console.log(store.getState())

)

// 发起一系列 action

store.dispatch(addTodo('Learn about actions'))

store.dispatch(addTodo('Learn about reducers'))

store.dispatch(addTodo('Learn about store'))

store.dispatch(toggleTodo(0))

store.dispatch(toggleTodo(1))

store.dispatch(setVisibilityFilter(VisibilityFilters.SHOW\_COMPLETED))

// 停止监听 state 更新

unsubscribe();

可以看到 store 里的 state 是如何变化的：