<https://react-redux.js.org/using-react-redux/static-typing>

Jest – js test runner allows you access of DOM via jsdom.

React Testing Library –

1. Snapshot testing – it is from jest and allows you to save snapshot of data with toMatchSnapshot() / toMatchInlineSnapshot(). This type of testing is used to check

Changes to view/ UI are expected or unexpected.

It needs **react-test-renderer** which uses ReactDOM internally.

React-test-renderer – it is package that provides experimental react renderer that can be used

To render react components to pure js objects, without depending on DOM

To update snapshot use npm run update

To test with snapshot testing use npm test

TestUtils are from jest and used to perform DOM testing .

More simplest way to perform DOM testing is using enzyme.

It has peer dependency on react react-dom

To use this go for below installations

Npm install –save-dev enzyme enzyme-adapter-react-16

Enzyme is js testing utility to test react components in more simplest way.

Enzyme-adapter – abstracts away anything that changes based on react version.

Redux is an implementation of flux. In place of dispatcher redux uses reducer.

Flux allows to have multiple stores if multiple states need to be maintained.

Redux does not allow multiple stores, it has centralized store to maintain state tree, it means redux store multiple state in centralized store in a form of tree.

Redux way of storing state tree is much better than flux.

Redux is described with 3 principles.

1. Single source of truth – global state of your application is stored in a single store in a form of object tree. Gives faster development as state is persisted in centralized store.
2. State is read only / immutable – state changes are possible with event handling and changes will happen to copy of it. Original state will remain as it is.
3. Changes are made with pure functions – i.e. reducers – it is a pure function that receives prev state and action, returns next state. It return new state object, instead of modifying previous state.

You can use single reducer and as the application grows reducer is broken into smaller reducers that will manage specific/respective part of state tree.

Redux uses

Redux-logger – it is middleware – used to log on console whenever state changes

Redux-promise-middleware – enables simple and robust handling of asynchronous action creators in redux

Redux-thunk – it is middleware – with plain basic store you can only do simple, synchronous updates but for asynchronous call you need this middleware, it can also handle complex synchronous calls.

Redux api is small and has only few methods

Store.getState() – returns current state of object tree

Store.dispatch() – dispatch action for change to state

Store.Subscribe(listener) – listen to changes in state tree

createStore function from redux is used to configure store.

It takes 3 arguments

1. Reducer
2. State initial value – optional
3. Enhancer – like middleware – optional

React-redux – it is a binding of react with redux. Here react component can read the data from redux store.

Npm install react-redux

Provider – ensures that the store from redux is available to complete application

Connect() – react-redux provides this function to connect react components to redux store

It takes below argument

mapDispatchToProps – used to dispatch action to store

mapStateToProps – react components can receive data from store using this