A store is an immutable object tree in Redux. A store is a state container which holds the application’s state. Redux can have only a single store in your application. Whenever a store is created in Redux, you need to specify the reducer.

Let us see how we can create a store using the **createStore** method from Redux. One need to import the createStore package from the Redux library that supports the store creation process as shown below −

import { createStore } from 'redux';

import reducer from './reducers/reducer'

const store = createStore(reducer);

A createStore function can have three arguments. The following is the syntax −

createStore(reducer, [preloadedState], [enhancer])

A reducer is a function that returns the next state of app. A preloadedState is an optional argument and is the initial state of your app. An enhancer is also an optional argument. It will help you enhance store with third-party capabilities.

A store has three important methods as given below −

getState

It helps you retrieve the current state of your Redux store.

The syntax for getState is as follows −

store.getState()

dispatch

It allows you to dispatch an action to change a state in your application.

The syntax for dispatch is as follows −

store.dispatch({type:'ITEMS\_REQUEST'})

subscribe

It helps you register a callback that Redux store will call when an action has been dispatched. As soon as the Redux state has been updated, the view will re-render automatically.

The syntax for dispatch is as follows −

store.subscribe(()=>{ console.log(store.getState());})

Note that subscribe function returns a function for unsubscribing the listener. To unsubscribe the listener, we can use the below code −

const unsubscribe = store.subscribe(()=>{console.log(store.getState());});

unsubscribe();