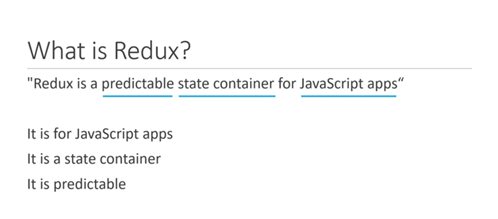
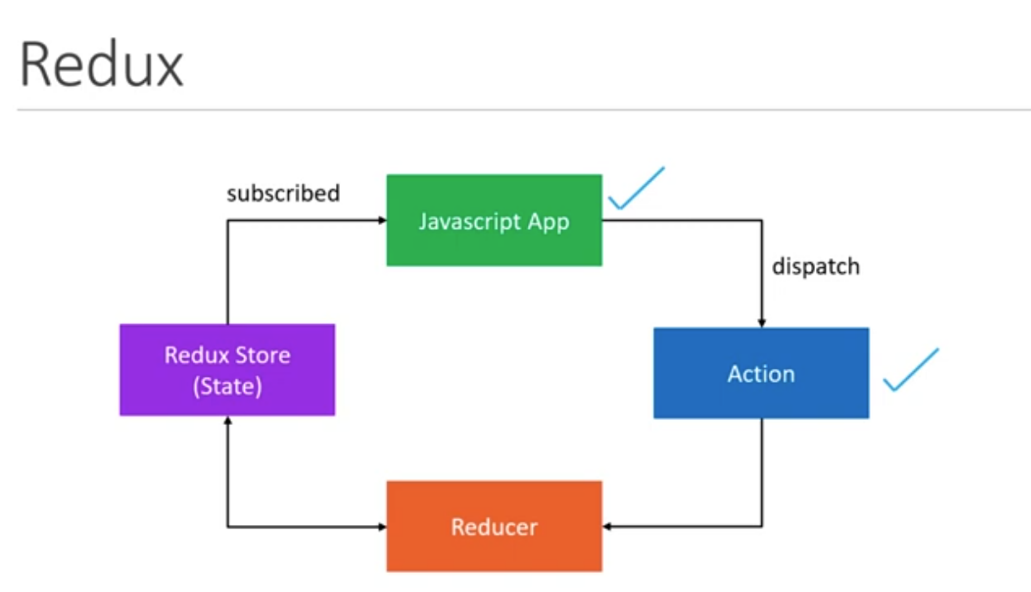
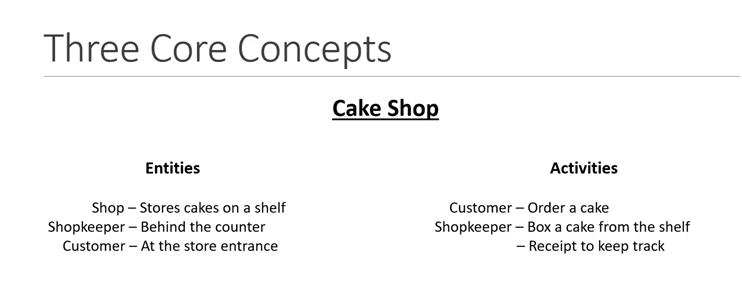
Redux in React

What is Redux?

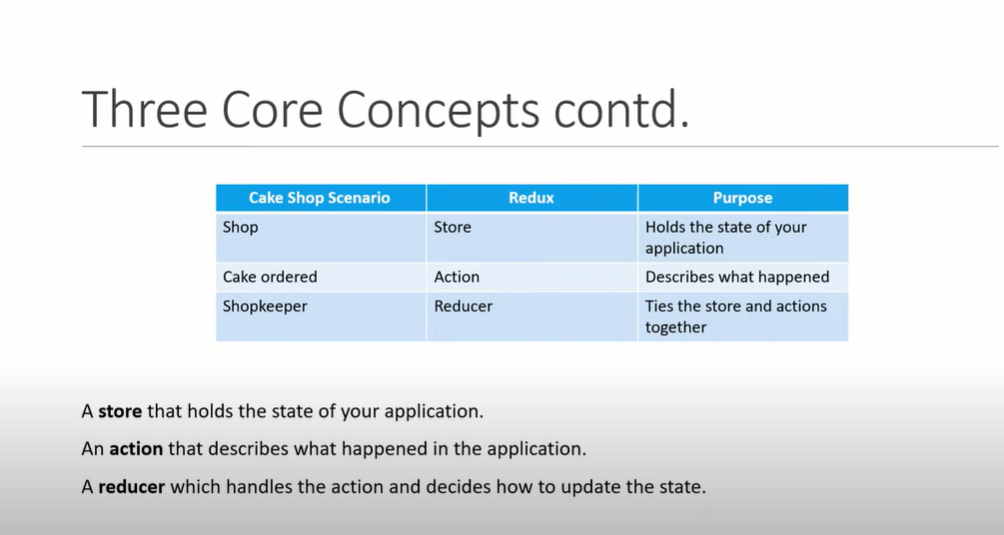


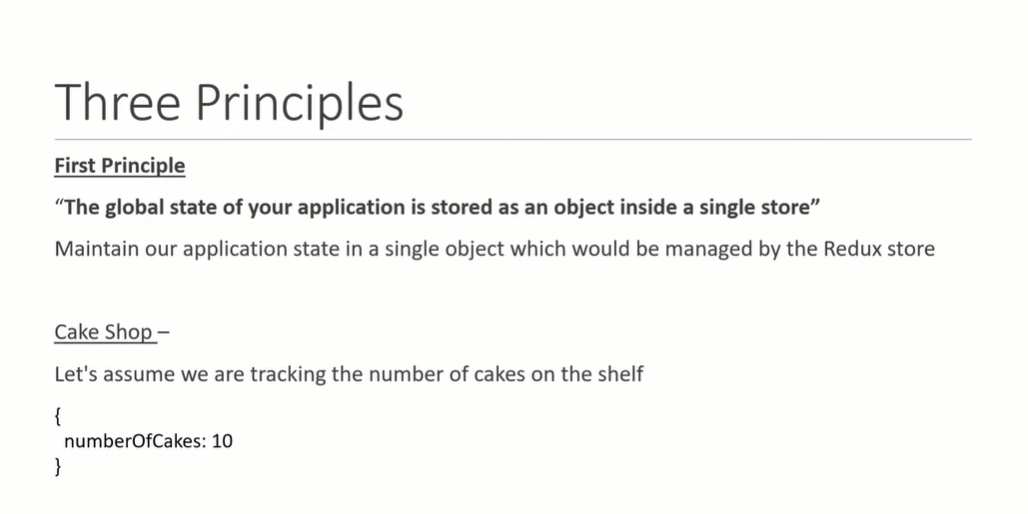
Three core concepts

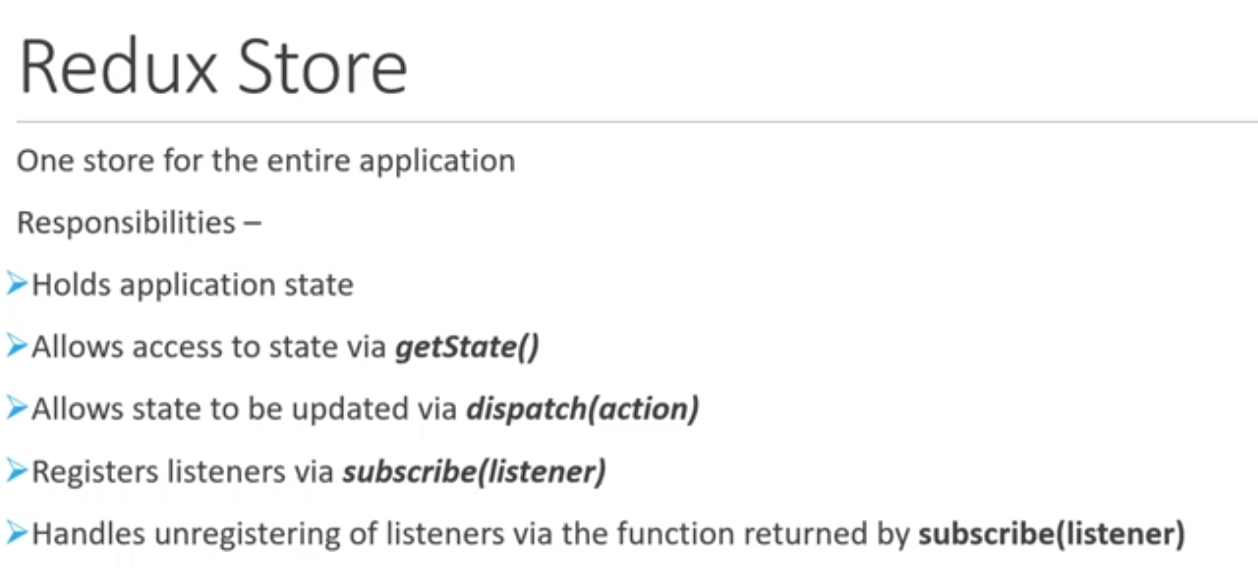


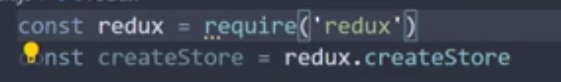


**Cake shop scenario**







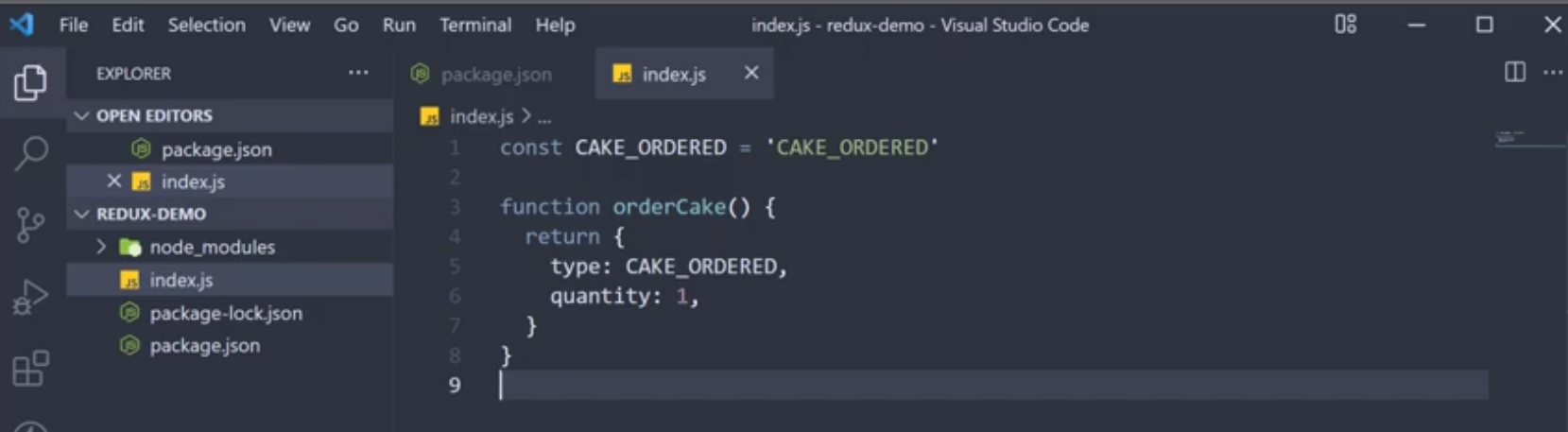




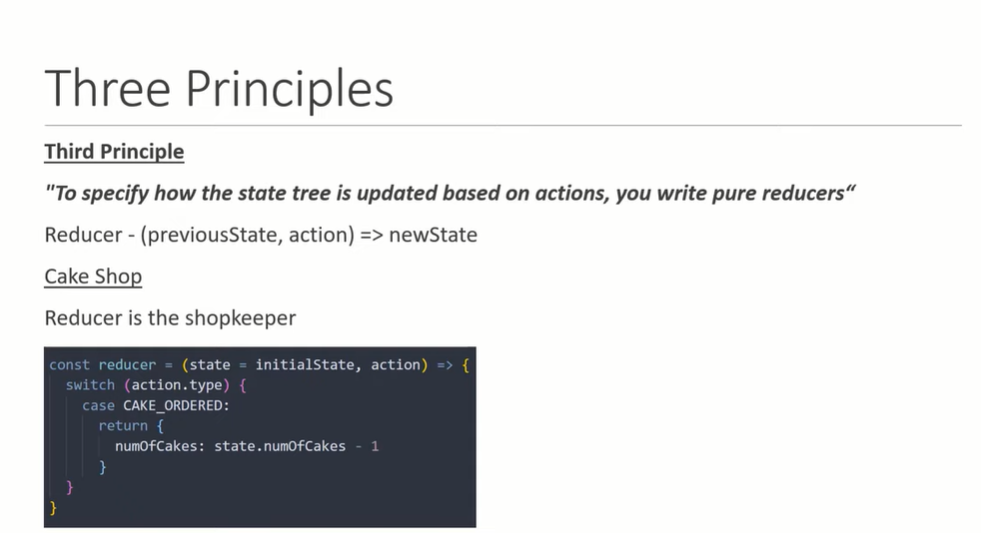
2.actions



Action Creator is function that returns an object.



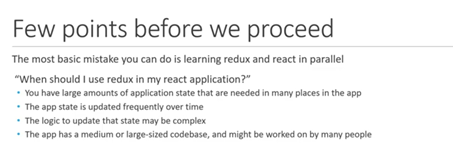
#.Reducers







When to use?



Create a fodler named reduxdemofirst

open the folder in VSC

npm init --yes in terminal

npm install redux

add new file index.js

node index.js

In index.js ad the code for actions, reducers and store and subscribe to store an dispatch actions to see how state gets ,odified.

E:\01\_SeedReactJS\ReduxToolkitDemos\redux\_demo

Implementing React-Redux

react redux library

npx create-react-app react-redux-demo

npm install redux react-redux

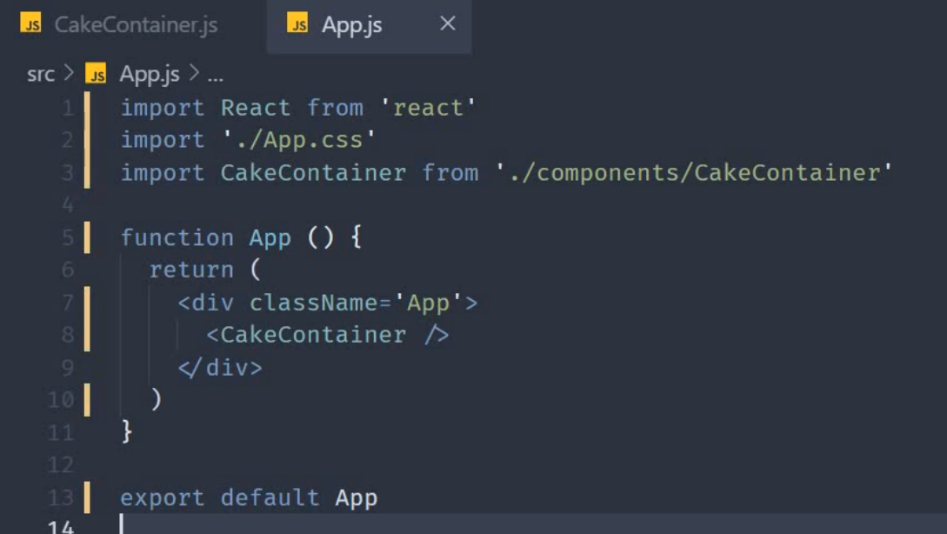
create componenets folder under "src"

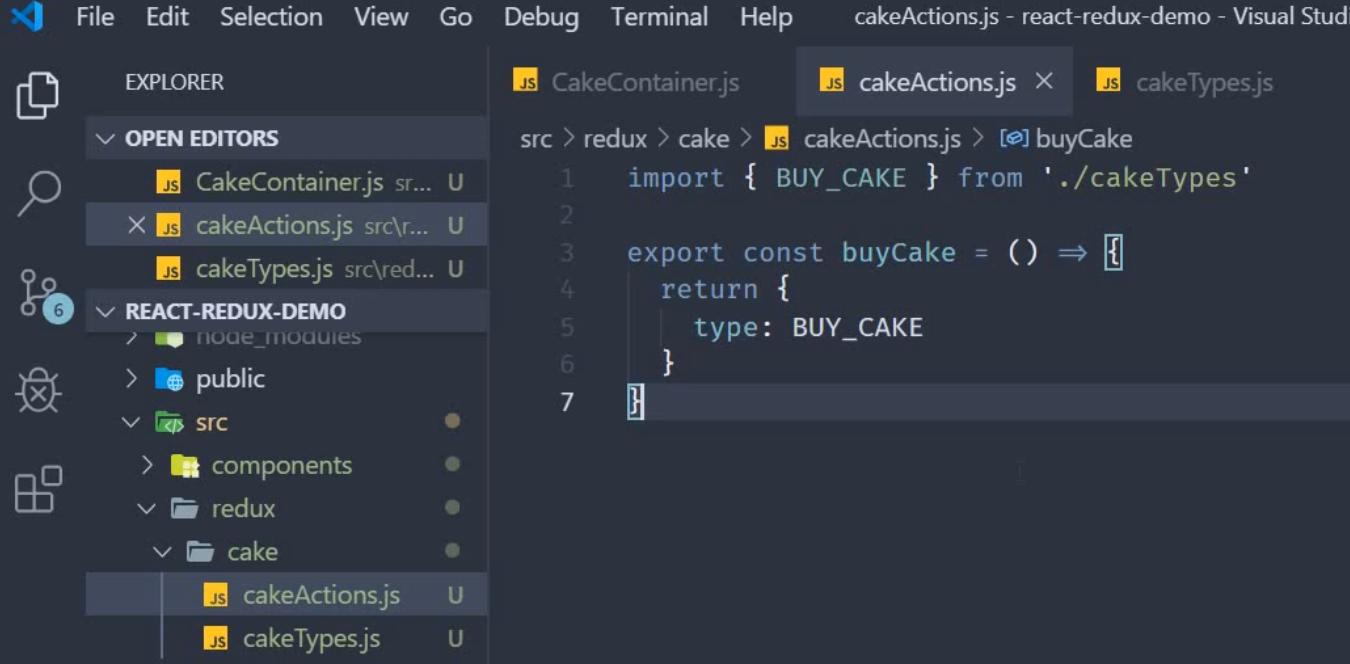
in it create CakeContainer.js

uf rfce

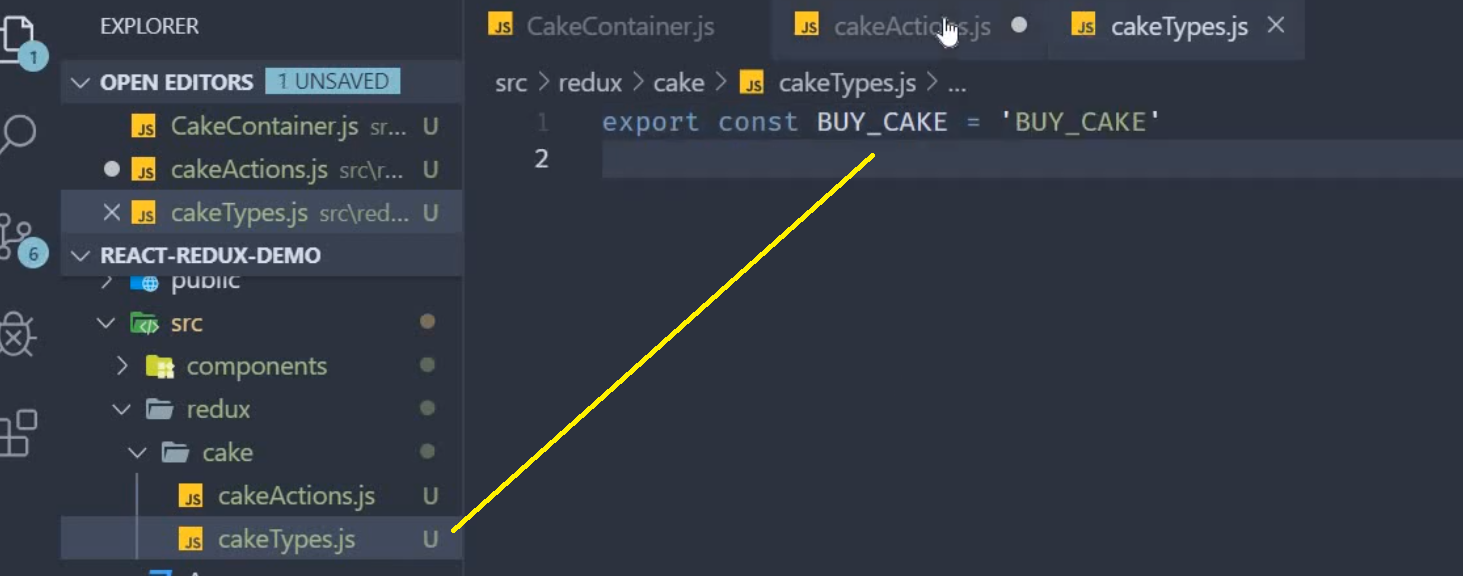
E:\01\_SeedReactJS\ReduxToolkitDemos\react-redux-demo



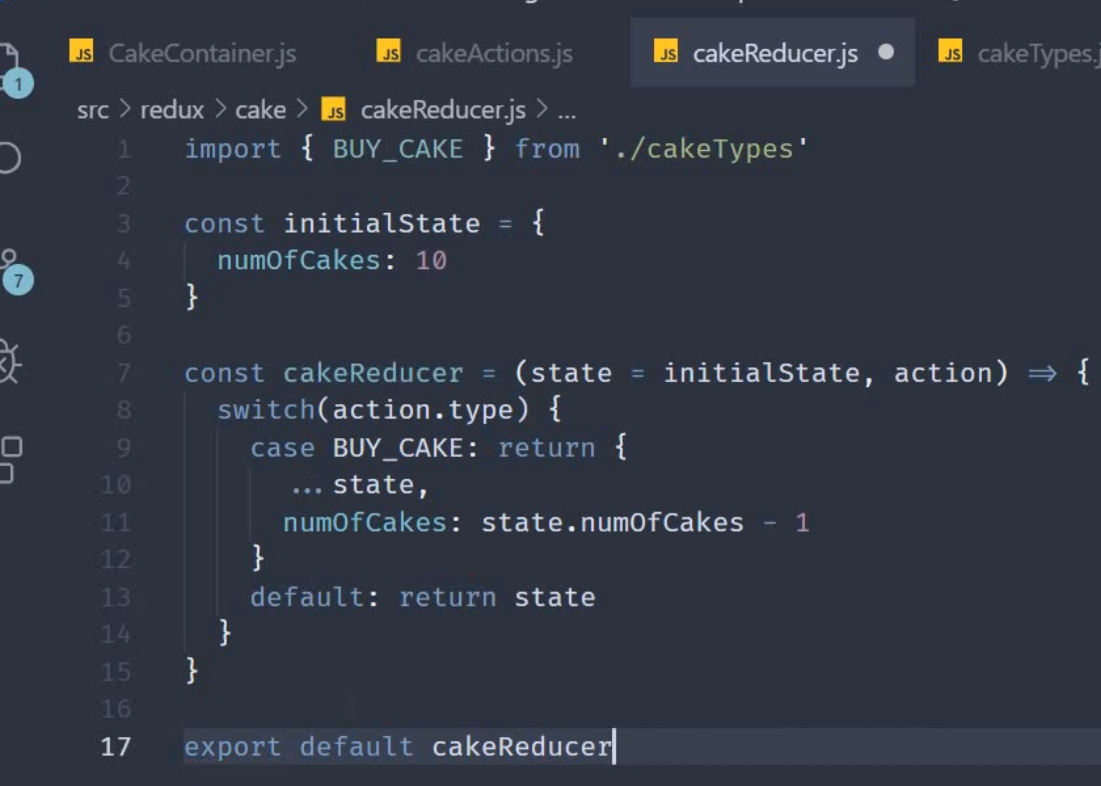




Create componenets. Redux, store fodler in src



Create cakeReducer.js in cake folder

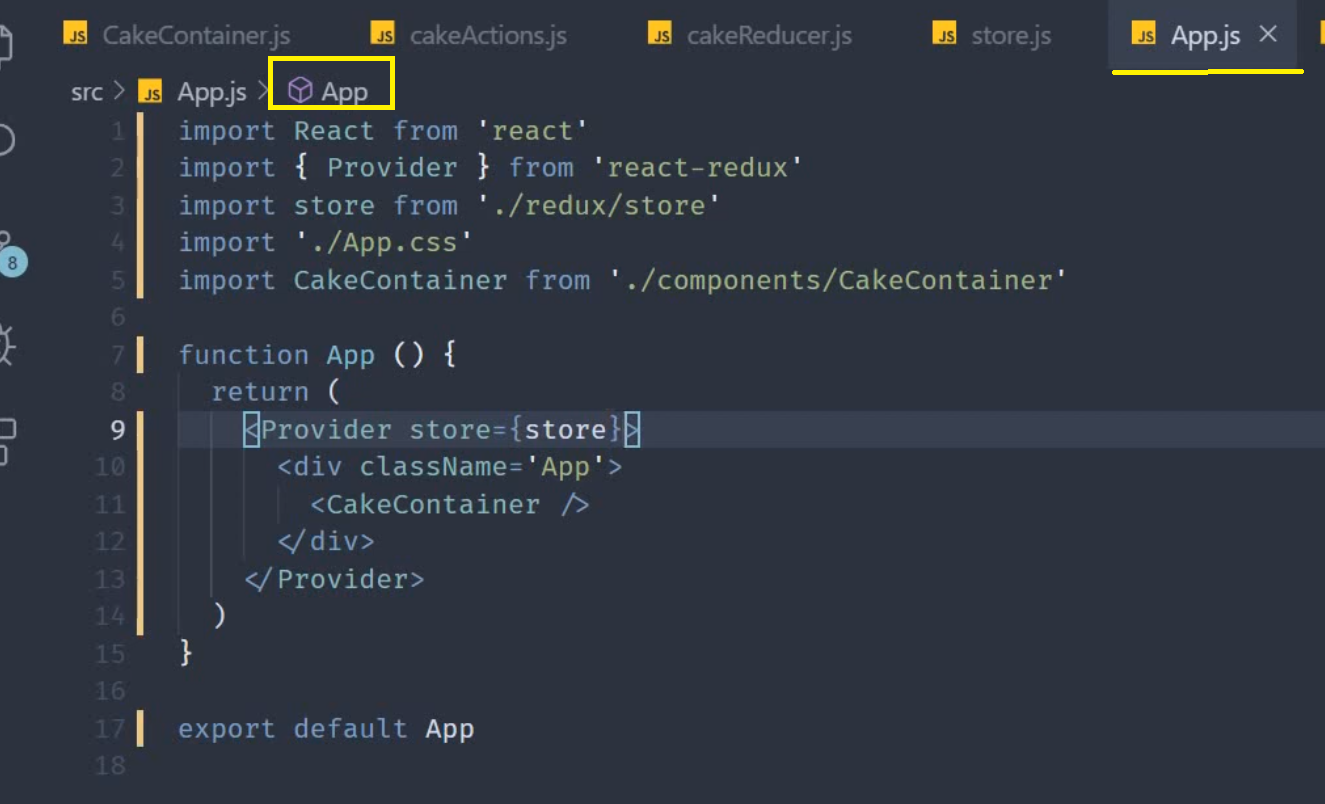


**In the redux folder create store.js**



Then go to App.js and provider store

To provider redux store to react application import provider from react-redux



Finally connect all pieces

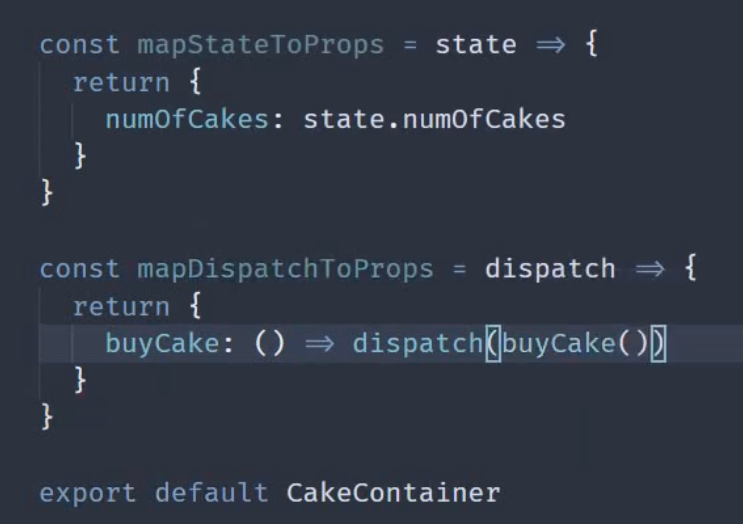
mapStateToProps() uses the redux state as a parameter and returns an object

mapDispatcToProps= takes dispatch() method as prop and returns an object.

In our application we have just one action creator.

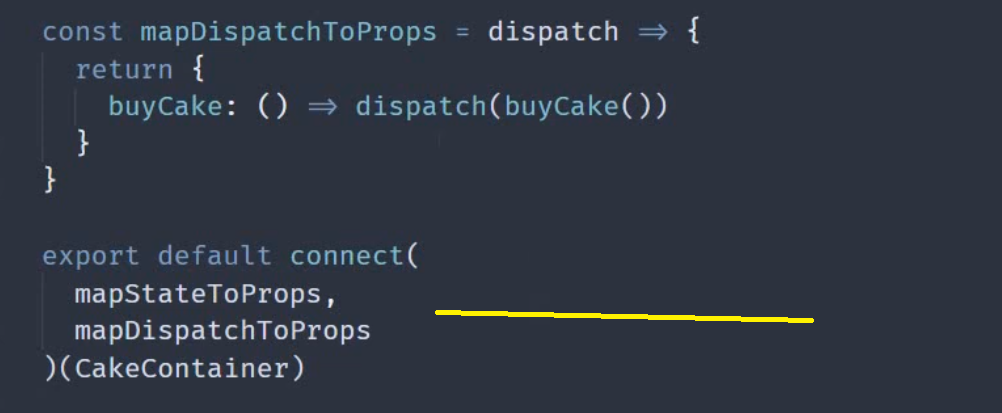
This function has one property buyCake: that dispatches actioncreator from redux

In cakecontainer.js



At the top in In cakecontainer.js

Import {connect} from react-redux

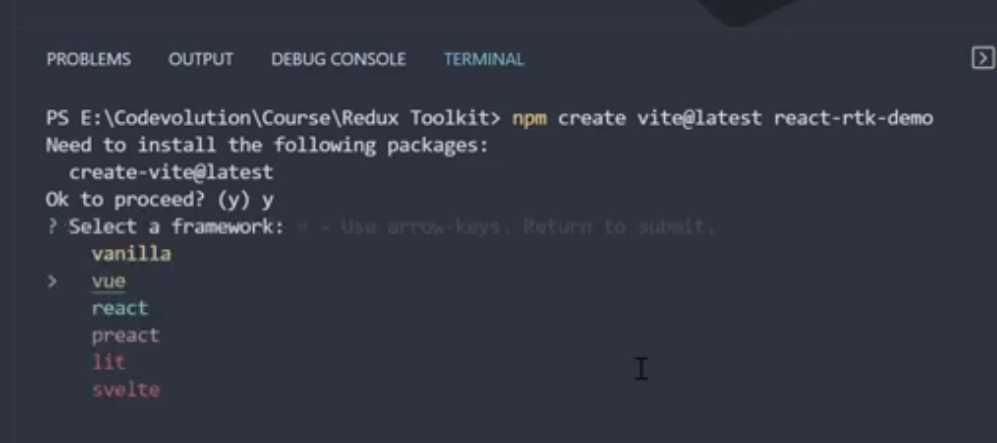


E:\01\_SeedReactJS\ReduxToolkitDemos\react-redux-demo (above demo)

React Reduc Toolkit with Vite tool

E:\01\_SeedReactJS\ReduxToolkitDemos\New folder\demo\_two\_rtk\_react

Using VITE and react redux toolkit



E:\01\_SeedReactJS\ReduxToolkitDemos>npm create vite@latest reatc-rtk-demo

Need to install the following packages:

create-vite@4.3.1

Ok to proceed? (y) y

√ Select a framework: » React

√ Select a variant: » JavaScript

Scaffolding project in E:\01\_SeedReactJS\ReduxToolkitDemos\reatc-rtk-demo...

Done. Now run:

cd reatc-rtk-demo

npm install

npm run dev

E:\01\_SeedReactJS\ReduxToolkitDemos> cd reatc-rtk-demo

E:\01\_SeedReactJS\ReduxToolkitDemos> reatc-rtk-demo >npm install

useSelector –hook is used get hold of the state

useDispatch-is used ti dispatch an action

useSelector hook is used to read data from the store & useDispatch hook is used to dispatch or trigger an event.

# teps To Follow

1. Create a Redux Store
2. Provide the Redux Store to React Main App.js
3. Create a Redux State Slice
4. Add Slice Reducers to the Store
5. Use Redux State and Actions in React Component
6. To use redux state in other component use useSelector hook from react-redux.

building a simple counter application which has an increment button and a decrement button. Redux toolkit is just a toolset for efficient redux development. It is intended to be the standard way to write Redux logic, and the redux team strongly recommend that you use it.

**Folder Structure.**

I have added an extra folder in the src directory called store and added two files. Before we dive into the nitty-gritties of redux toolkit, lets define some words that you will encounter as far as redux is concerned.

**Store.**

This is a state container which holds the application's state.

**Reducers**.

A reducer function receives the state of an app and an action. Based on the action, it will calculate the next state and return the new state.

***Slice***

It is a collection of reducers for a single feature in an app. In our case, we are going to create a slice for the counter feature and we are going to call it counterSlice

**Creating The Counter Slice**

First things first, install react-redux and redux toolkit npm i react-redux @reduxjs/toolkit

We are going to create the slice in reducer.js file in the store folder.

To create the slice, we have to import **createSlice** from redux toolkit.

import {createSlice} from "@reduxjs/toolkit"

Now lets define our initial state.

const initialState= {counter:0}

The next step is to create the slice.

export const counterSlice= createSlice(

{

name:"counter",

initialState,

reducers:{

increment:(state)=>{

state.counter+=1

},

decrement: (state)=>{

state.counter-=1

}

}

}

)

createSlice is a function which takes in an object.

It has a name property where you can provide a string value of your choice.

The next property is the initialState property whose value is the initialState we created above. Notice how we don't have to write it as initialState:initialState because the key has the same name as the value.

The next property is the reducer which contains all the reducer functions.

Each reducer function has a state parameter which is just equal to the initialState.

The increment function returns state.counter+=1 while the decrement function returns state.counter-=1.

Next, we have to export the two reducers by using counterSlice.actions

export const {increment, decrement}= counterSlice.actions

Lastly, export export default counterSlice.reducer

**ConfigureStore**

Now in index.js still in the store folder, we have to create our store.

We do this by importing configureStore from redux toolkit.

import { configureStore } from "@reduxjs/toolkit";

We also need to import the counterSlice.reducer from the reducer.js file.

import counterReducer from "./reducer"

configureStore is a function that takes in an object as a parameter.

One property available is called **reducer** whose value is the **slice.reducer**

const store= configureStore({

reducer:{

counter: counterReducer

}

})

export default store

Now navigate to index.js and import Provider from redux toolkit. Its kind of similar to the provider available in the context Api.

import {Provider} from "react-redux"

Now we have to wrap it around our App component so that the store is accessible through out App.

import React from 'react';

import ReactDOM from 'react-dom';

import './index.css';

import App from './App';

import {Provider} from "react-redux"

import store from "./store/index"

ReactDOM.render(

<React.StrictMode>

<Provider store={store}>

<App />

</Provider>

</React.StrictMode>,

document.getElementById('root')

);

We have to pass the store to the provider. <Provider store={store}>

Now lets navigate to App.js.

We have to import useSelector and useDispatch from react-redux.

import {useSelector,useDispatch} from 'react-redux'

useSelector will allow us to extract data from the store.

useDispatch hook returns a reference to the dispatch function from the Redux store. we use it to dispatch actions as needed.

We have to import increment and decrement functions.

import {increment, decrement} from './store/reducer'

Lets get the initial value of counter.

const counter= useSelector((value)=>value.counter.counter)

Now lets get the dispatch function.

const dispatch= useDispatch()

We then have to create two functions which will be executed when the increment and decrement buttons are clicked.

const dispatch= useDispatch() const handleIncrement= ()=>{ dispatch(increment()) } const handleDecrement= ()=>{ dispatch(decrement()) }

Below is the full code for App.js

import {useSelector,useDispatch} from 'react-redux'

import {increment, decrement} from './store/reducer'

function App() {

const counter= useSelector((counter)=>counter.counter.counter)

const dispatch= useDispatch()

const handleIncrement= ()=>{

dispatch(increment())

}

const handleDecrement= ()=>{

dispatch(decrement())

}

return(

<>

<h1>COUNTER</h1>

<h4>{counter}</h4>

<button onClick={handleIncrement}>Increment</button>

<button onClick={handleDecrement}>Decrement</button>

</>

);

}

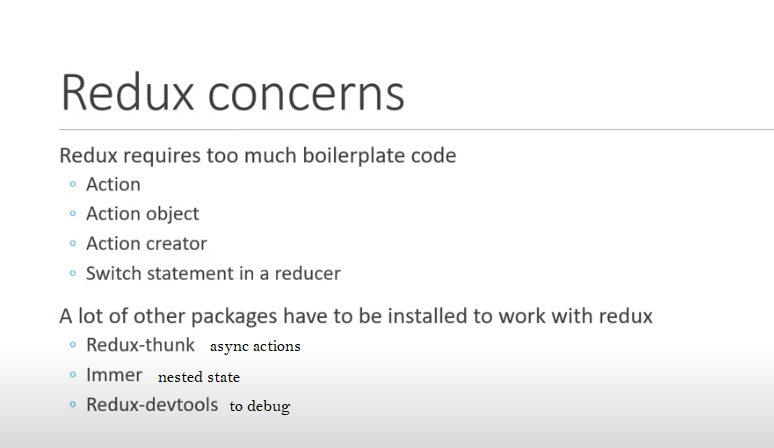
export default App;

Now run npm start and your counter App should be working.

Redux Toolkit

Why toolkit?





What is RDK?

