**Lesson 04 Demo 04**

**React-combine-reducer-example**

**Objective:** To demonstrate the react with redux to combine two reducer in one store to do the task depending upon the reducer.

**Tools required:** Node JS and React JS

**Prerequisites:** HTML, CSS, JavaScript ES5/ES6, Basic React Concept

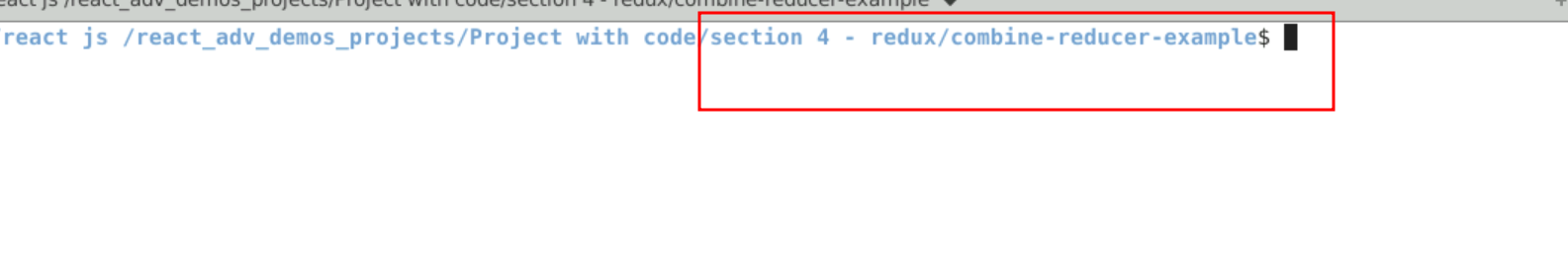
**Note** : All react js project already created with version 18.x with Sample App.js file

**Steps to be followed:**

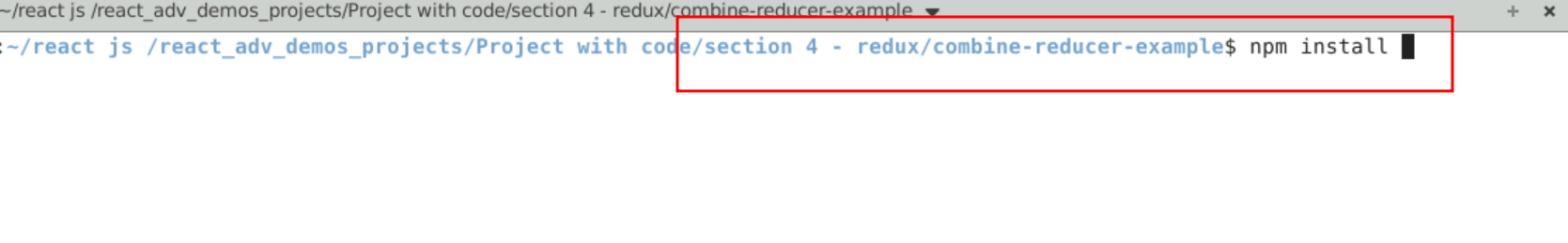
1. Set up for react js project
2. Create action folder which contains index.js file which provide the action for reducer.
3. Create reducer folder which contains more than one file ie couter.js reducer, isLogged.js reducer and index.js which combine both reducer.
4. In App.js file use both reducer with help of useSelector and useDispatch hook.
5. In index.js file configure the store details.
6. In index.html part of public folder add the boostrap details.
7. Test the application.

**Step 1: Set up for react js project**

1. Open a terminal window inside a React JS pre-created project ie **combine-reducer-example**

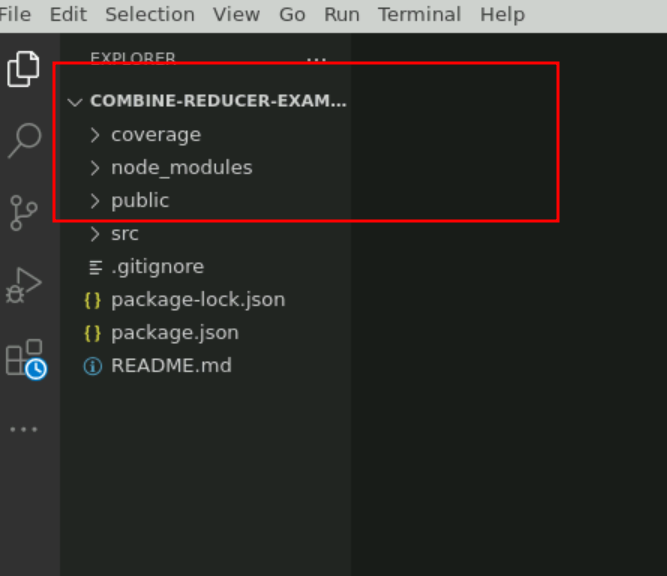


1. Now you need to run the command as **npm install.** This command helps us to installed all required dependencies mention in package.json file in local machine in the form of node\_module folder.

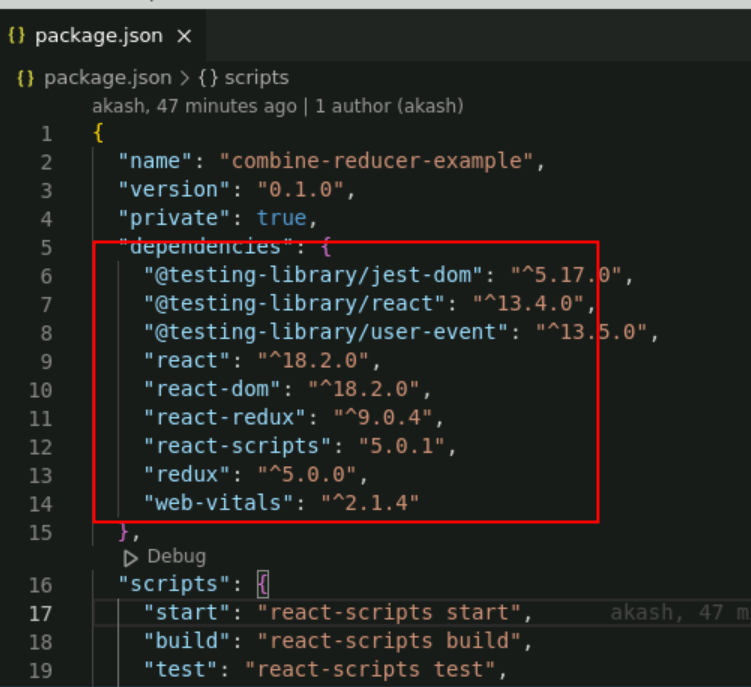


1. Now open **combine-reducer-example** folder in VS Code Editor

Note: short cut to open write **code .**

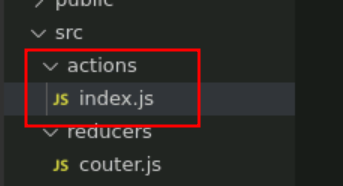
****

1.4 now open package.json file and view external dependencies.

****

**Step 2: Create action folder which contains index.js file which provide the action for reducer.**

2.1 create the action folder and create the index.js file which contains the action for reducer.



2.2 index.js file contains all action details

//this function will be used by our /reducers/counter.js file

export const increment = (num) => {

return {

//we could name type key anything we want but we shouldnt

type: 'INCREMENT',

payload: num

}

}

export const decrement = () => {

return {

type: 'DECREMENT'

}

}

export const handleLoggin = () => {

return{

type: "SIGN\_IN"

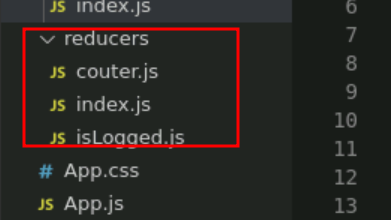
}

}

**Step 3 : Create reducer folder which contains more than one file ie couter.js reducer, isLogged.js reducer and index.js which combine both reducer.**

.

3.1 reducer folder with couter.js, isLogged.js and index.js file



3.2 couter.js file which contains reducer function which takes state and action and based upon it will do some operation of state variable.

**couter.js**

// counter REDUCER

//takes care of all the action related to our counter

// this will take 2 argument

const counterReducer = (state = 0, action) => {

switch(action.type){

case 'INCREMENT':

return state + action.payload

case 'DECREMENT':

return state - 1

default:

return state

}

}

export default counterReducer

3.3 isLogged.js file which contains reducer function which takes state and action and based upon action it will do some operation of state variable.

**isLogged.js**

const loggedReducer = (state=false, action) =>{

switch(action.type){

case 'SIGN\_IN':

return !state;

default:

return state;

}

}

export default loggedReducer

**3.4** index.js file. This file combine more than one reducer.

Index.js

import counterReducer from './couter'

import loggedReducer from './isLogged'

import {combineReducers} from 'redux'

const allReducers = combineReducers({

//we access this by any key we want

counter: counterReducer,

isLogged: loggedReducer

})

export default allReducers

**Step 4 : In App.js file use both reducer with help of useSelector and useDispatch hook.**

**App.js**

import React, { useState } from 'react';

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

import {increment,decrement,handleLoggin} from './actions'

import './App.css';

function App() {

//next line will allow us to grab the state inside counter. So it goes to /reducers/index.js

//find the counter key in allReducers function, which gets it from /reducers/counter.js where the

//state and counterReducer function reside, now depends on the action type being used

let [buttonValue,setButtonValue]=useState("Login");

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

const isLogged = useSelector(state => state.isLogged)

const dispatch = useDispatch()

return (

<div className="container">

<h1>Counter {counter}</h1>

<button onClick={()=> dispatch(increment(5))} disabled={!isLogged} className='btn btn-success'>+</button>

<button onClick={()=> dispatch(decrement())} disabled={!isLogged} className='btn btn-success'>-</button>

<div>

<input type="button" onClick={()=> {

dispatch(handleLoggin())

isLogged ?setButtonValue("Login"):setButtonValue("Logout")

}} value={buttonValue} className='btn btn-primary'/>

</div>

{isLogged ? <div><h1>Only show when loggin is true</h1></div> : ''}

</div>

);

}

export default App;

**Step 5 : In index.js file configure the store details.**

5.1 In index.js file please configure store details.

**index.js file**

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import App from './App';

import reportWebVitals from './reportWebVitals';

**import {legacy\_createStore as createStore} from 'redux'**

**import allReducer from './reducers'**

**import { Provider } from 'react-redux'**

**const store = createStore(allReducer);**

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

<React.StrictMode>

**<Provider store={store}>**

**<App />**

**</Provider>**

</React.StrictMode>

);

// If you want to start measuring performance in your app, pass a function

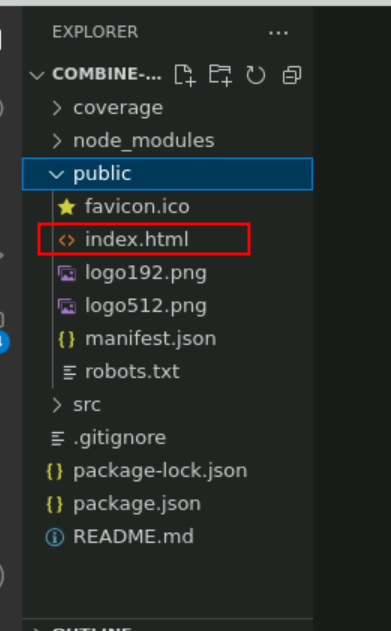
// to log results (for example: reportWebVitals(console.log))

// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals

reportWebVitals();

**Step 6 : In index.html part of public folder add the boostrap details.**

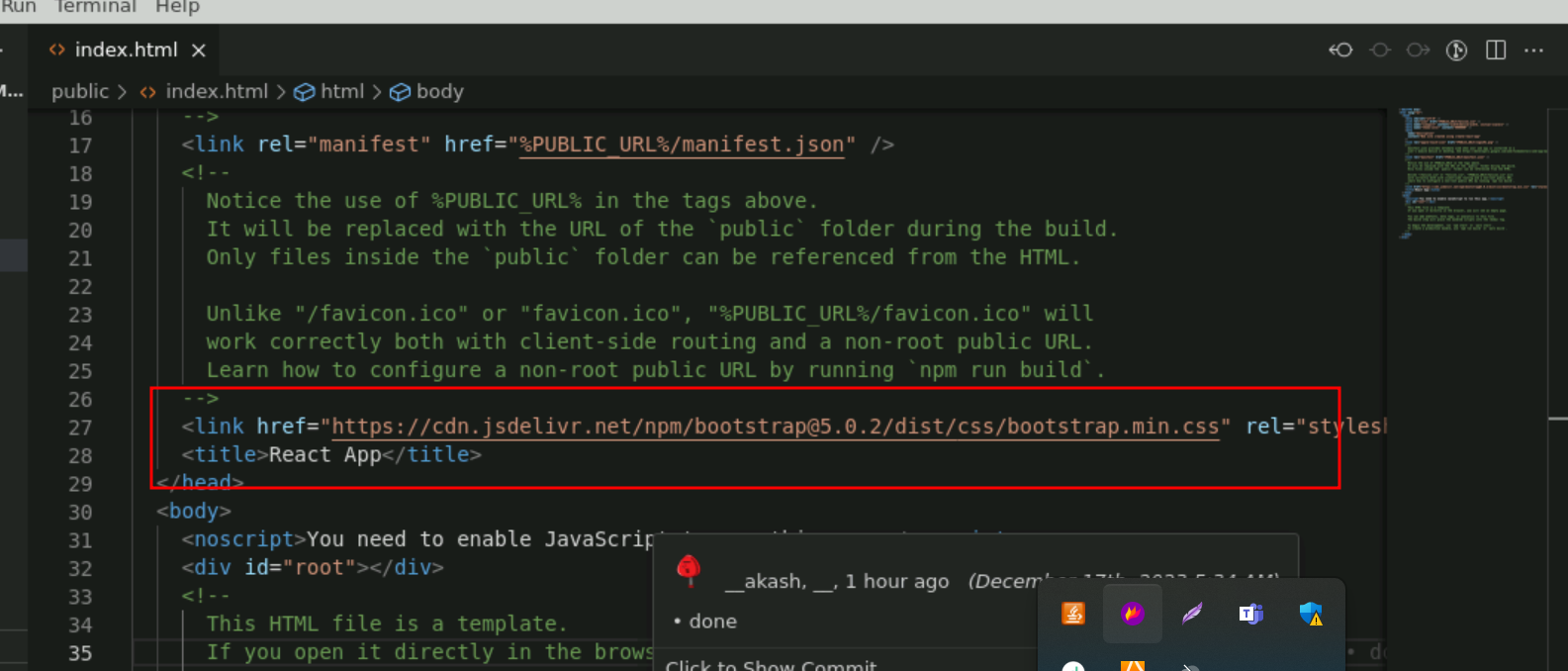
6.1 index.html file

****

6.2 in index.html file copy and paste the bootstap url

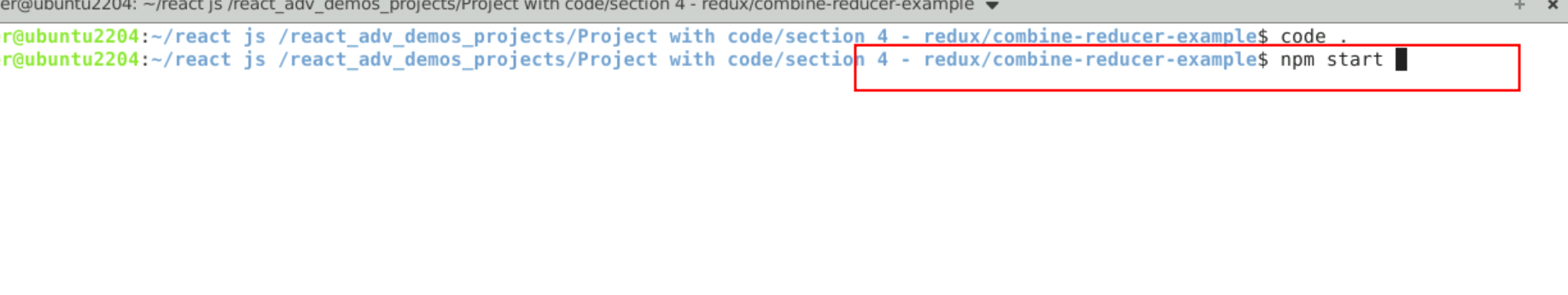
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

index.html

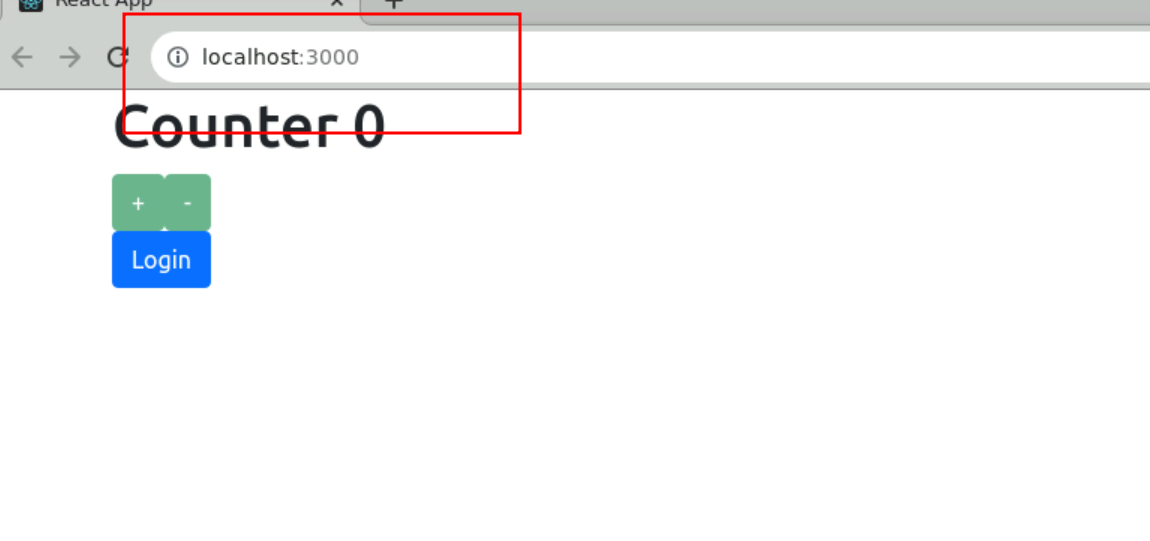


**Step 7 : Now we run the application using npm start**

7.1 open the terminal and run the command as **npm start**

****

7.2 Now you can view the output on browser.



7.3 after click on login button then only increment and decrement button enable to do the increment and decrement values.

