**Lesson 04 Demo 02**

**React with Redux counter app**

**Objective:** To demonstrate the react with redux to access counter value in each component and do the operation on counter variable like increment and decrement.

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

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

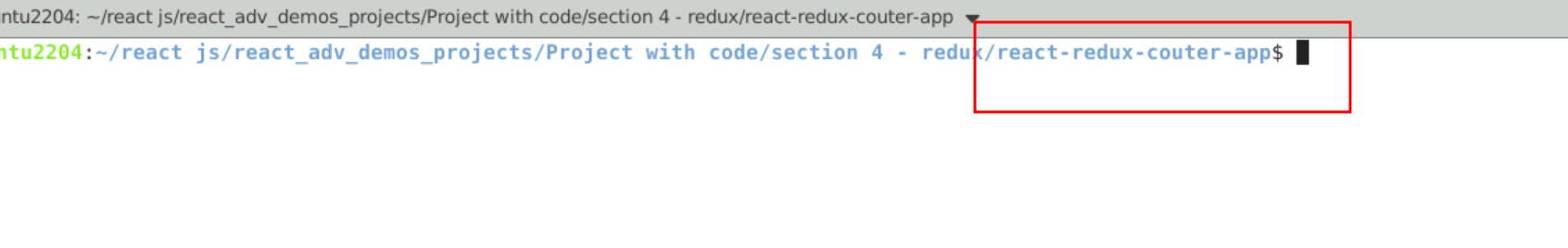
**Note** : All react js project already created with version 18.x with Sample App.js file

**Steps to be followed:**

1. Set up for react js project
2. Create reducer, store and configuration for store in index.js function.
3. Create user defined component to access global state variable as well as create Increment and Decrement component to change the global state variable value.
4. In App.js file import all user defined component and test the application.

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

1. Open a terminal window inside a React JS pre-created project ie **react-redux-gs-variable**

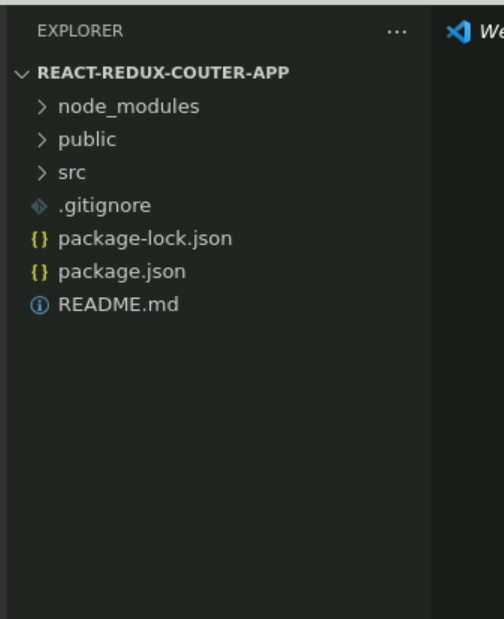


1. Now you need to run the command as **npm install.** This command helps us to installed all required dependencies mention in package.json file in local machine in the form of node\_module folder.

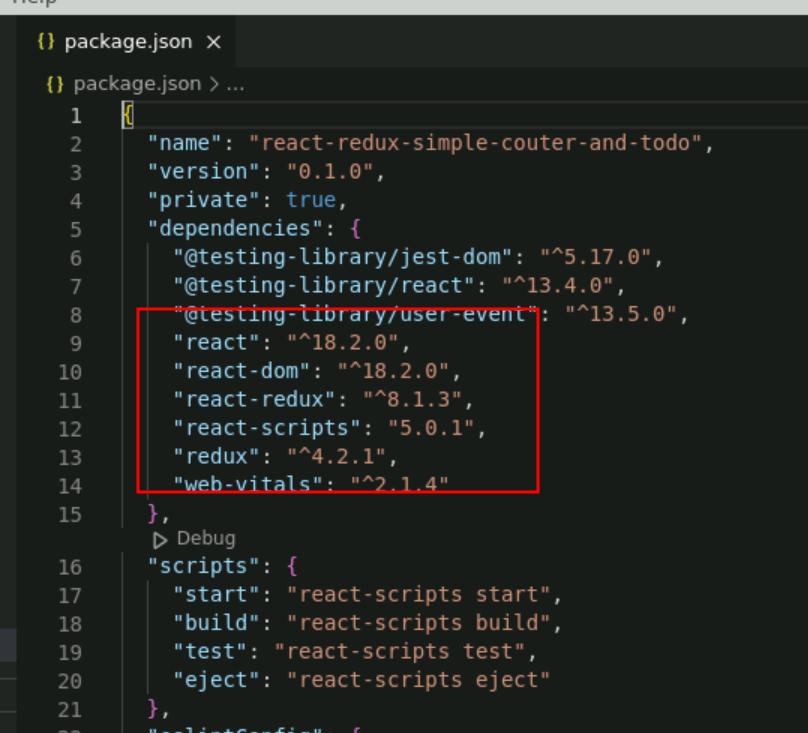


1. Now open **react-redux-counter-app** folder in VS Code Editor

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



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

****

**Step 2:** Create reducer, store and configuration for store in index.js function.

2.1 create counterReducer.js file. Which contains normal JavaScript function which takes two parameter ie state and action. Base upon action it will do some operation on state variable. And these variable as consider as global variable if we configure with store.

// Action Type

const INCREMENT = 'INCREMENT';

const DECREMENT = 'DECREMENT';

// Reducer

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

switch (action.type) {

case INCREMENT:

return state + 1;

case DECREMENT:

return state - 1;

default:

return state;

}

};

**2.2** We need to create react store with help of redux library. Here store connect with reducer to make state variable as global variable.

couterStore.js

// Store

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

import { counterReducer } from './couterReducer';

export const couterStore = createStore(counterReducer);s

**2.3 Now we need to configure store the index.js file. Inside this file we need to take the help of Provider part of react-redux and connect to store. Provider must apply for whole application so wrap for App root components.**

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import App from './App';

import reportWebVitals from './reportWebVitals';

**import {couterStore} from './couterStore';**

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

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

root.render(

<React.StrictMode>

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

<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 3 :** Create user defined component to access global state variable as well as create Increment and Decrement component to change the global state variable value.

.

3.1 create counter.js file. This component with help of useSelector hook it can access global state variable

import {useSelector} from 'react-redux';

export const Couter = ()=> {

const data = useSelector(state => state);

return(

<div>

<h3>Counter Component</h3>

<p>Counter value in Counter component {data}</p>

</div>

)

}

3.2 Now create IncrementComponent.js file. Which take the help of useDispatch() function to increment action on state variable with help of reducer.

import {useDispatch} from 'react-redux';

// Action Type

const INCREMENT = 'INCREMENT';

// Actions

const increment = () => ({ type: INCREMENT });

export const IncrementCounter = ()=> {

const dispatch = useDispatch();

return(

<div>

<h3>Increment Component</h3>

<input type="button" value="Increment" onClick={()=>dispatch(increment())}/>

</div>

)

}

3.3 Now create DecrementComponent.js file. Which take the help of useDispatch() function to decrement action on state variable with help of reducer.

import {useDispatch} from 'react-redux';

// Action Type

const DECREMENT = 'DECREMENT';

// Actions

const decrement = () => ({ type: DECREMENT });

export const DecrementCounter = ()=> {

const dispatch = useDispatch();

return(

<div>

<h3>Decrement Component</h3>

<input type="button" value="Decrement" onClick={()=>dispatch(decrement())}/>

</div>

)

}

**Step 4.** In App.js file import all user defined component and test the application.

**4.1** In App.js file import Counter, IncrementCounter and DecrementCounter to test the application.

import './App.css';

import { Couter } from './Couter';

import { DecrementCounter } from './DecrementCounter';

import { IncrementCounter } from './IncrementCounter';

function App() {

return (

<div className="App">

<Couter></Couter>

<hr/>

<IncrementCounter></IncrementCounter>

<DecrementCounter></DecrementCounter>

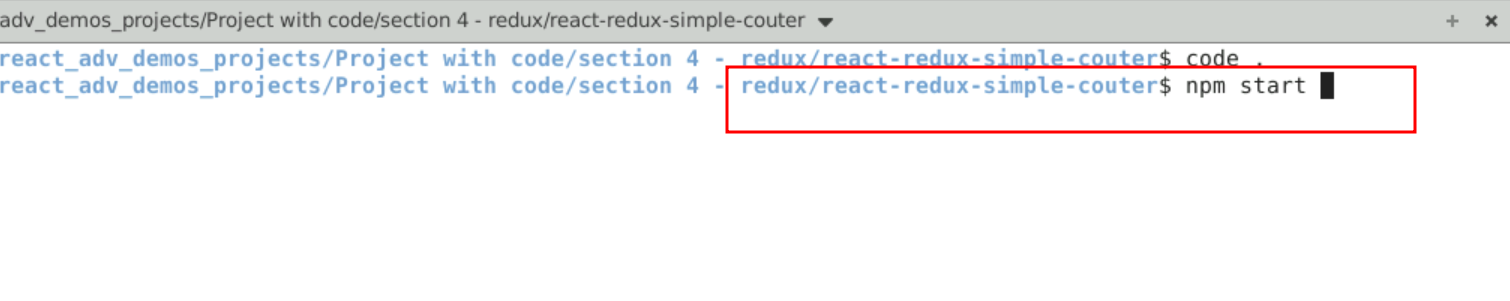
</div>

);

}

export default App;

**4.2 Now we run the application using npm start**

****

**4.3** Now you can view the output on browser. You can view counter value through counter component and you can increment and decrement counter value from another component.

