

REDUX COUNTER

Dr. Jose Annunziato

Import all necessary dependencies

Create a new react application

```
$ create-react-app simple-react-redux-counter
```

```
$ cd simple-react-redux-counter
```

Install **redux**

```
$ npm install redux --save
```

Install **react-redux**

```
$ npm install react-redux --save
```

In index.js, import all necessary dependencies

```
import React, { Component } from 'react'
```

```
import ReactDOM from 'react-dom'
```

Let's create a simple counter application

const *Counter* = ({value = 0}) =>

<div>

****{value}****

<button> + **</button>**

<button> - **</button>**

</div>

```
ReactDOM.render(  
  <Counter/>,  
  document.getElementById('root')  
)
```



Add onClick event handlers to the buttons

```
<button onClick={e => {  
  alert('plus')  
}}> + </button>
```

```
<button onClick={e => {  
  alert('minus')  
}}> - </button>
```

Dispatch actions onClick

```
const Counter = ({value = 0, dispatch}) =>
```

```
  <div>
```

```
    <span>{value}</span>
```

```
    <button onClick={e => {  
      dispatch({type: 'plus'})  
    }}> + </button>
```

```
    <button onClick={e => {  
      dispatch({type: 'minus'})  
    }}> - </button>
```

```
  </div>
```

Create a reducer to handle the plus action

```
const counterReducer =  
  (state = { value: 0 }, action) => {  
    switch (action.type) {  
      case 'plus':  
        alert('plus')  
        return { value: ++state.value } ...  
    }  
  }
```


Also handle the minus action

```
const counterReducer =  
  (state = { value: 0 }, action) => { ...  
    case 'minus':  
      alert('minus')  
      return { value: --state.value }  
    default:  
      return state  
  }  
}
```

Connect actions, create and provide the store

```
const CounterApp = connect()(Counter)
const store = createStore(counterReducer)
```

```
ReactDOM.render(
  <Provider store={store}>
    <CounterApp/>
  </Provider>,
  document.getElementById('root')
)
```

Map state's value to component's value property

```
const CounterApp = connect(  
  state => ({value: state.value})  
)(Counter)
```

But it's better to encapsulate map into function

```
const stateToPropertyMapper = (state) => {  
  return {  
    value: state.value  
  }  
}
```

```
const CounterApp = connect  
(stateToPropertyMapper)(Counter)
```

Constants

Dr. Jose Annunziato

Declare constants in a separate file

In src/constants/index.js, declare constants

```
export const PLUS = 'plus'
```

```
export const MINUS = 'minus'
```

In src/index.js, import constants

```
import { PLUS, MINUS } from './constants/index'
```

Replace all constant literals

```
const Counter = ({value = 0, dispatch}) => {  
  return (  
    <div><span>{value}</span>  
      <button onClick={e => {  
        dispatch({type: PLUS})  
      }}> + </button>  
      <button onClick={e => {  
        dispatch({type: MINUS})  
      }}> - </button>  
    </div>  
  )  
}
```

Replace all constant literals

```
const counterReducer = (state = { value: 0 }, action) => {  
  switch (action.type) {  
    case PLUS:  
      return { value: ++state.value }  
    case MINUS:  
      return { value: --state.value }  
    default:  
      return state  
  }  
}
```


Action Creators

Dr. Jose Annunziato

Bind onClick to properties

```
const Counter = ({value = 0, plus, minus}) => {  
  return (  
    <div>  
      <span>{value}</span>  
      <button onClick={plus}> + </button>  
      <button onClick={minus}> - </button>  
    </div>  
  )  
}
```

Encapsulate actions into reusable functions

```
const plus = dispatch => (  
  dispatch({type: PLUS})  
)
```

```
const minus = dispatch => (  
  dispatch({type: MINUS})  
)
```

Create a dispatch to property mapper

```
const dispatcherToPropertyMapper  
  = dispatch => ({  
    plus: () => plus(dispatch),  
    minus: () => minus(dispatch)  
  })
```

Connect dispatch mapper

```
const CounterApp = connect(  
  stateToPropertyMapper,  
  dispatcherToPropertyMapper)  
  (Counter)
```

Break Up Into Separate Files

Dr. Jose Annunziato

Move action creators to src/actions/index.js

```
import {PLUS, MINUS} from '../constants/index'  
export const plus = dispatch => (  
  dispatch({type: PLUS}))  
export const minus = dispatch => (  
  dispatch({type: MINUS}))
```

From src/index.js, import actions

```
import { plus, minus } from './actions/index'
```

Move reducers to src/reducers/index.js

```
import { PLUS, MINUS } from '../constants/index'
const counterReducer =
  (state = { value: 0 }, action) => {
    switch (action.type) {
      case PLUS: return { value: ++state.value }
      case MINUS: return { value: --state.value }
      default: return state
    }
  }
export default counterReducer
```


Move components into src/components

```
import React from 'react'
```

```
const Counter = ({value = 0, plus, minus}) => {
```

```
  return (
```

```
    <div>
```

```
      <span>{value}</span>
```

```
      <button onClick={plus}> + </button>
```

```
      <button onClick={minus}> - </button>
```

```
    </div>
```

```
  )}
```

```
export default Counter
```

Move containers into src/containers

Move CounterApp.js into src/containers/CounterApp.js

Import all dependencies

```
import React from 'react'  
import Counter from '../components/Counter'  
import { plus, minus } from '../actions'  
import { connect } from 'react-redux'
```

Move containers into src/containers

Implement state and dispatch to property matchers

```
const stateToPropertyMapper = state => ({  
  value: state.value  
})
```

```
const dispatcherToPropertyMapper = dispatch => ({  
  plus: () => plus(dispatch),  
  minus: () => minus(dispatch)  
})
```

Connect and export the container

```
const CounterApp = connect(  
  stateToPropertyMapper,  
  dispatcherToPropertyMapper)  
  (Counter)
```

```
export default CounterApp
```

Final index.js

```
import React, { Component } from 'react'
import ReactDOM from 'react-dom'
import { createStore } from 'redux'
import { Provider } from 'react-redux'
import counterReducer from './reducers'
import CounterApp from './containers/CounterApp'
const store = createStore(counterReducer)
ReactDOM.render(
  <Provider store={store}>
    <CounterApp/>
  </Provider>,
  document.getElementById('root')
)
```