

Fig. 1: Reducer

Reducer can only change the state.

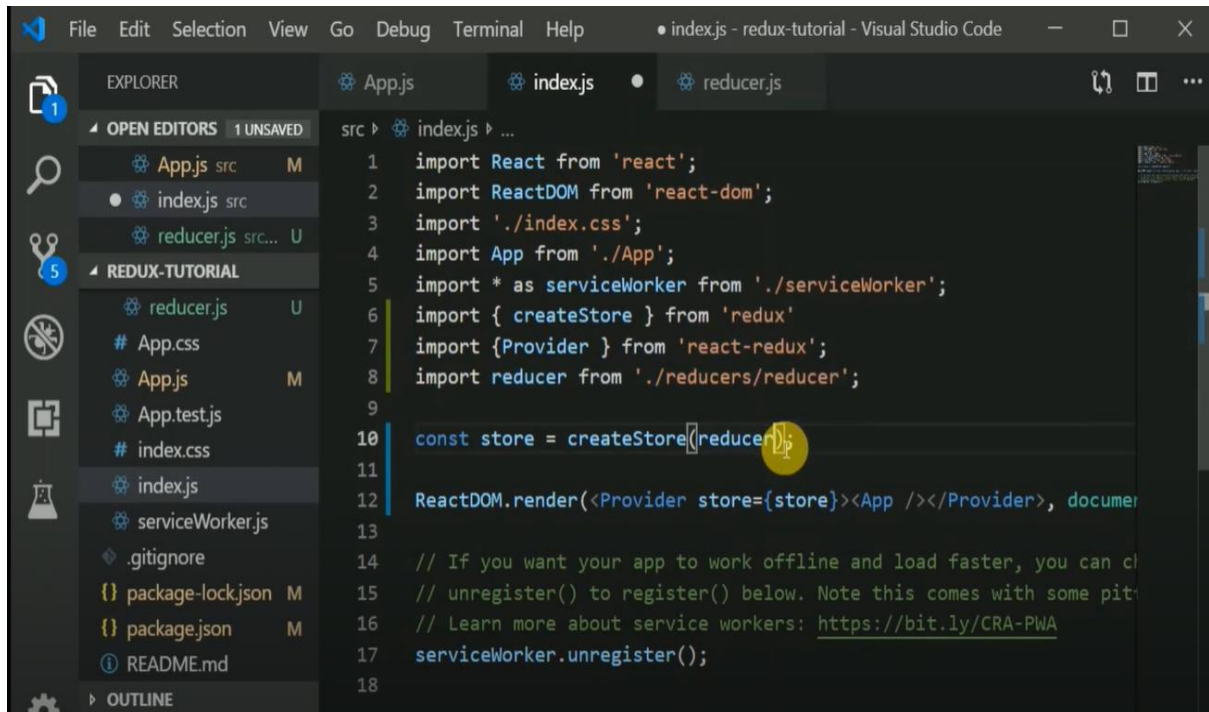


Fig. 2: Store

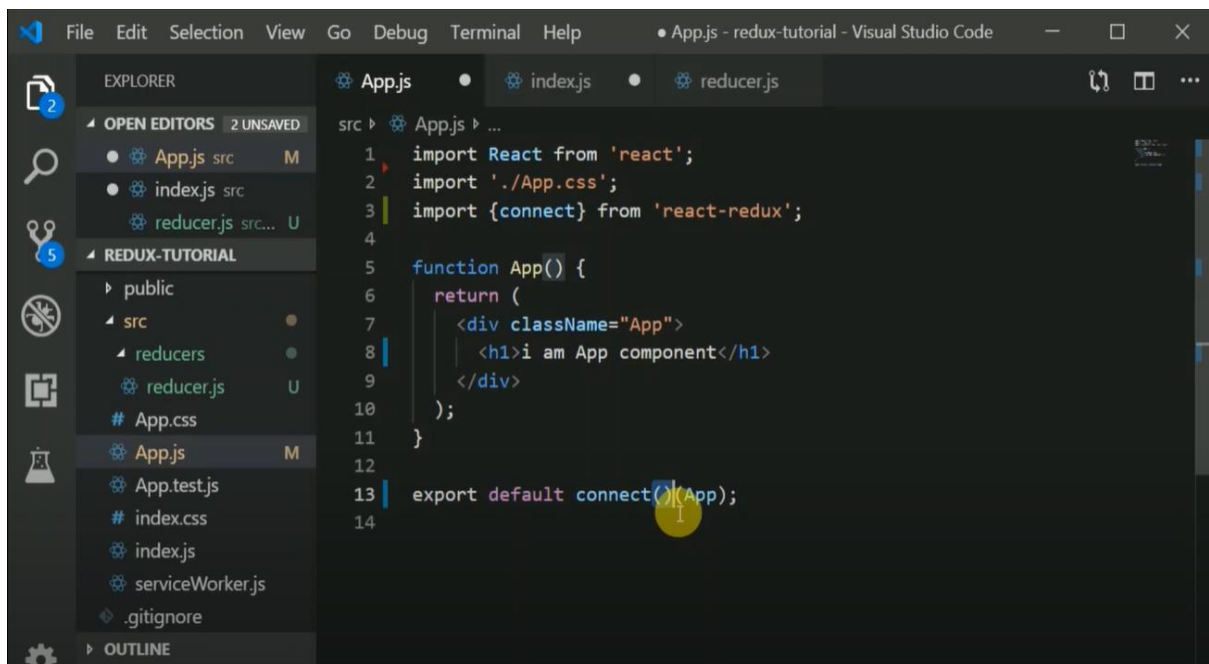


Fig. 3: Connect

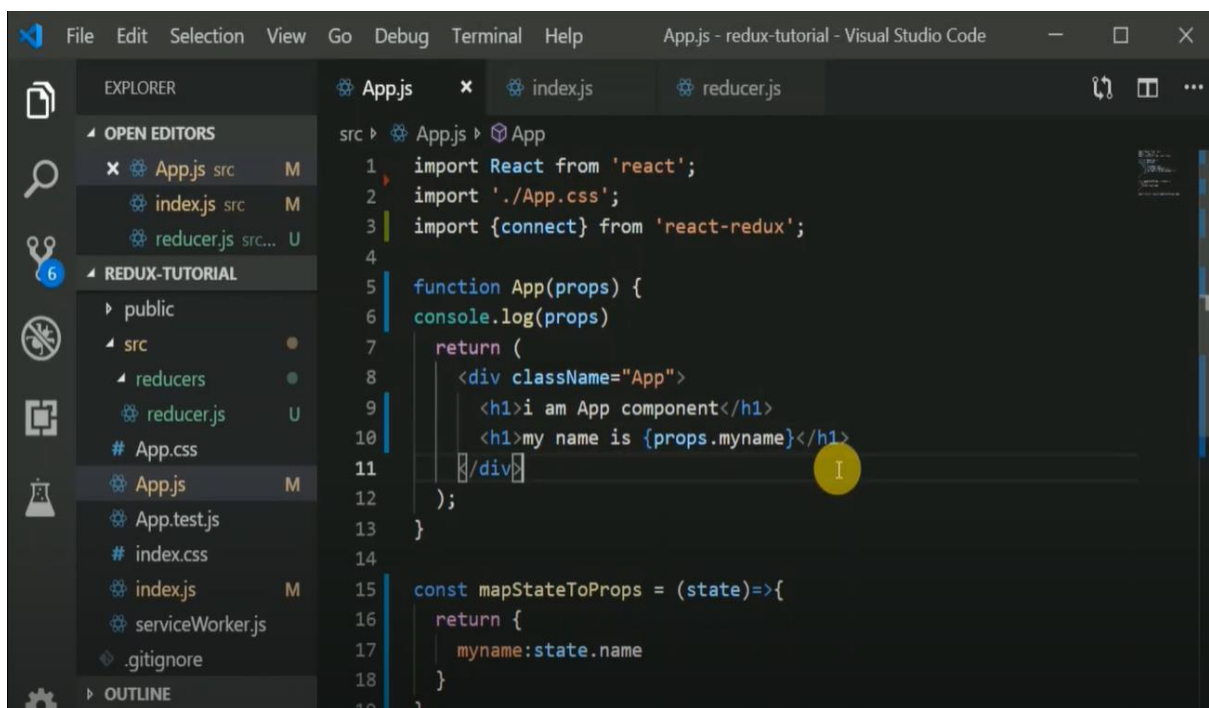
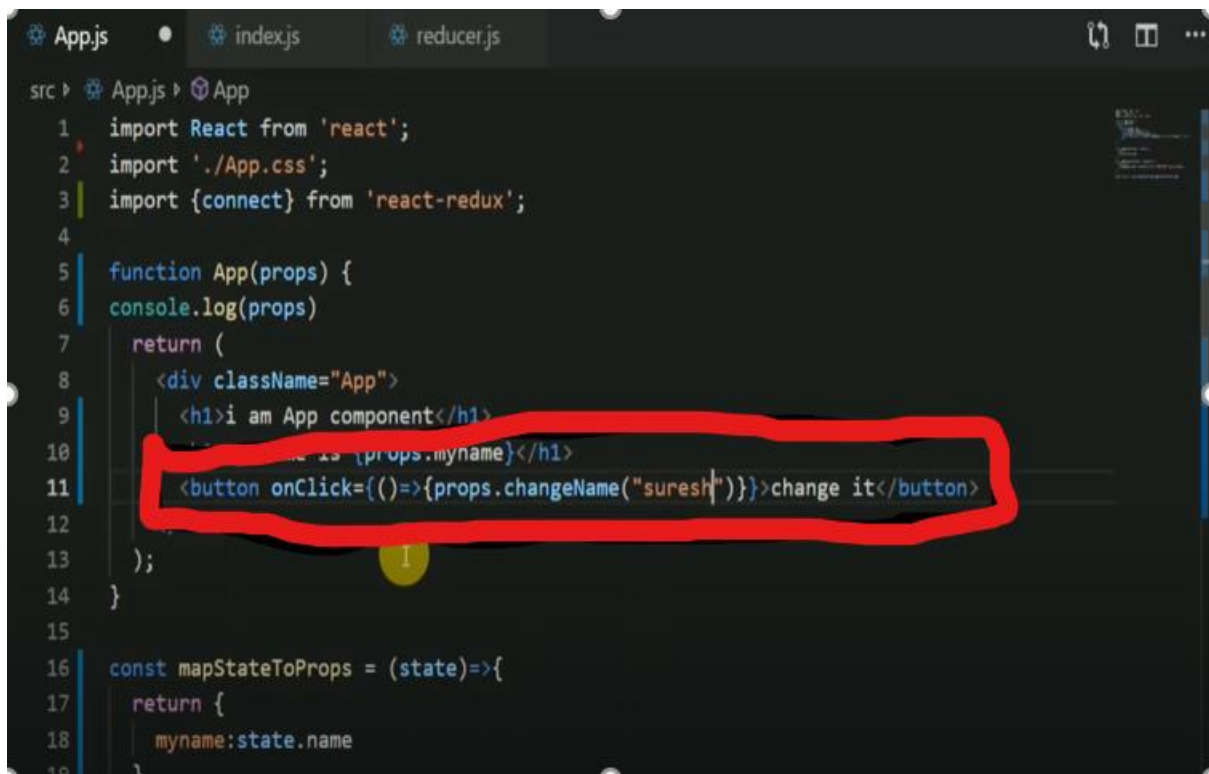


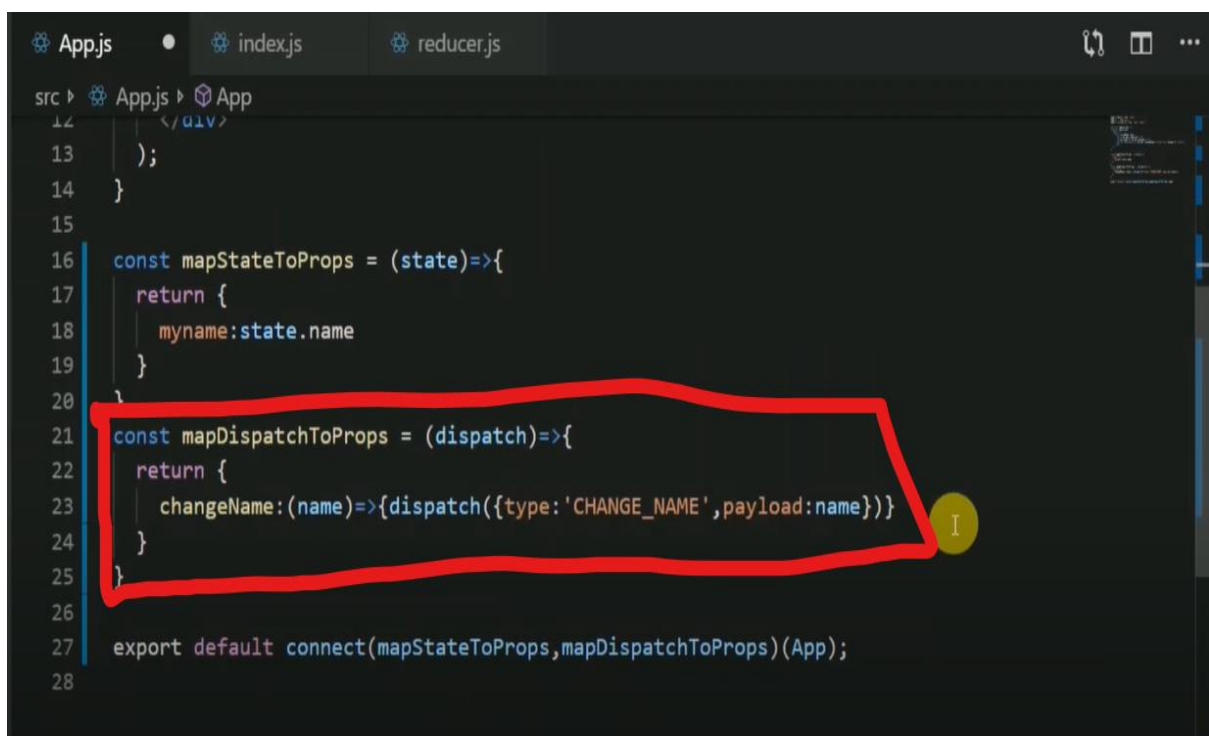
Fig. 4: Map state to props.

Map state to props is used to receive data coming from store.



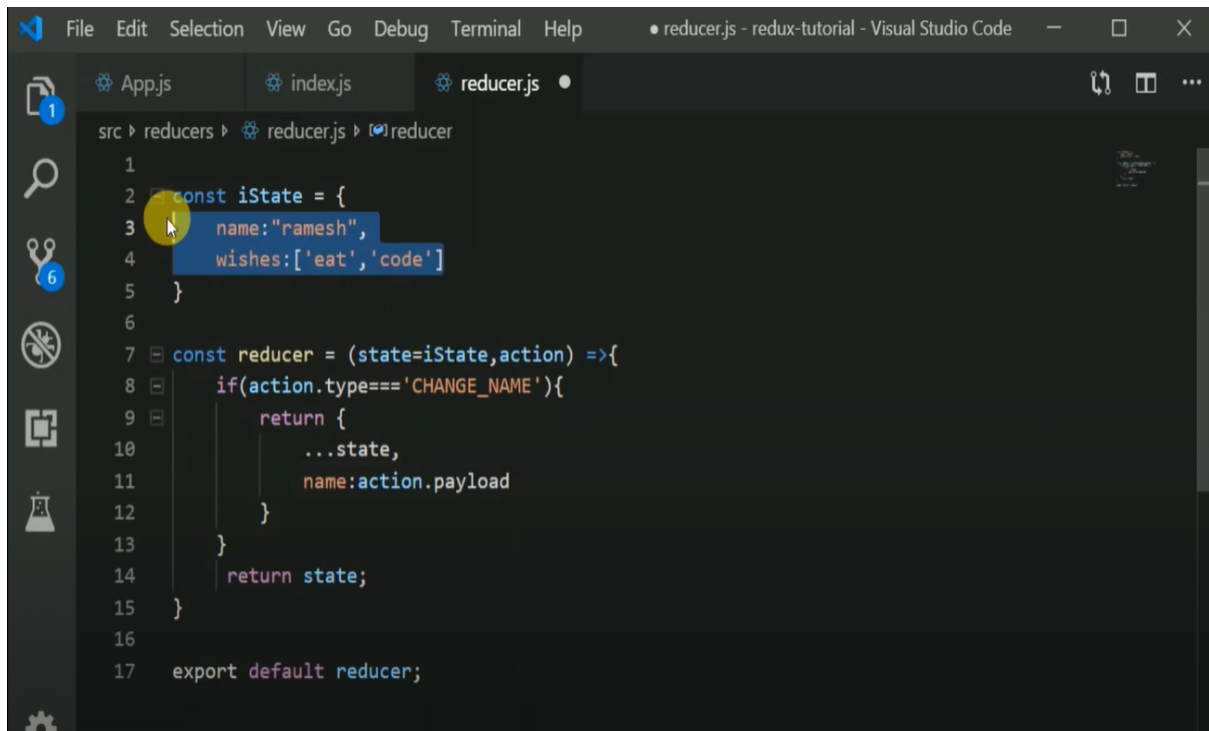
```
App.js • index.js reducer.js
src > App.js > App
1 import React from 'react';
2 import './App.css';
3 import {connect} from 'react-redux';
4
5 function App(props) {
6   console.log(props)
7   return (
8     <div className="App">
9       <h1>i am App component</h1>
10      <h2> {props.myname}</h2>
11      <button onClick={()=>{props.changeName("suresh")}}>change it</button>
12    </div>
13  );
14 }
15
16 const mapStateToProps = (state)=>{
17   return {
18     myname:state.name
19   }
20 }
```

Fig. 5: Calling and passing argument to the function (in reducer)



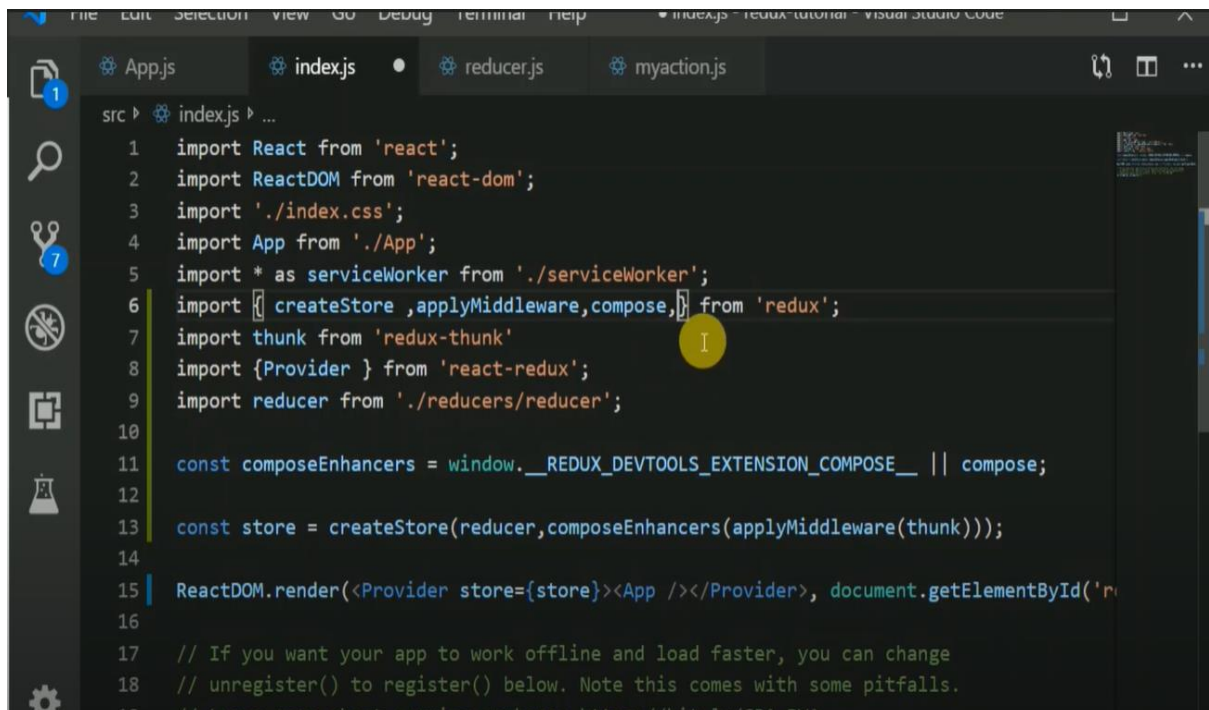
```
App.js • index.js reducer.js
src > App.js > App
12 </div>
13 );
14 }
15
16 const mapStateToProps = (state)=>{
17   return {
18     myname:state.name
19   }
20 }
21 const mapDispatchToProps = (dispatch)=>{
22   return {
23     changeName:(name)=>{dispatch({type:'CHANGE_NAME',payload:name})}
24   }
25 }
26
27 export default connect(mapStateToProps,mapDispatchToProps)(App);
28
```

Fig. 6: Defined function in App.js



```
1  const iState = {
2    name: "ramesh",
3    wishes: ['eat', 'code']
4  }
5
6
7  const reducer = (state=iState, action) =>{
8    if(action.type==='CHANGE_NAME'){
9      return {
10        ...state,
11        name: action.payload
12      }
13    }
14    return state;
15  }
16
17  export default reducer;
```

Fig. 7: Logic defined in reducer.



```
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3  import './index.css';
4  import App from './App';
5  import * as serviceWorker from './serviceWorker';
6  import { createStore, applyMiddleware, compose, } from 'redux';
7  import thunk from 'redux-thunk';
8  import { Provider } from 'react-redux';
9  import reducer from './reducers/reducer';
10
11  const composeEnhancers = window.__REDUX_DEVTOOLS_EXTENSION_COMPOSE__ || compose;
12
13  const store = createStore(reducer, composeEnhancers(applyMiddleware(thunk)));
14
15  ReactDOM.render(<Provider store={store}><App /></Provider>, document.getElementById('root'));
16
17  // If you want your app to work offline and load faster, you can change
18  // unregister() to register() below. Note this comes with some pitfalls.
19  // Learn more about service workers: https://bit.ly/CRA-PWA
```

Fig. 8: Store (II)

More than one reducer video link: [Combine reducer](#)