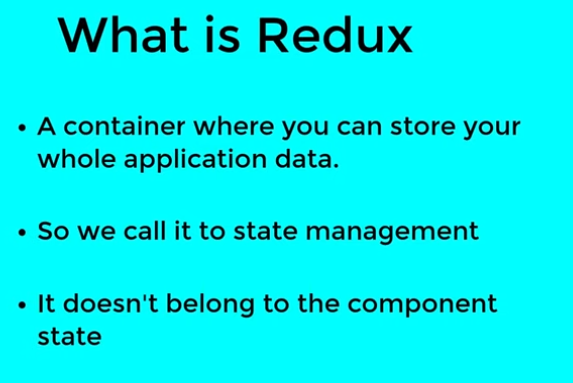


Redux is kind of big type ARRAY

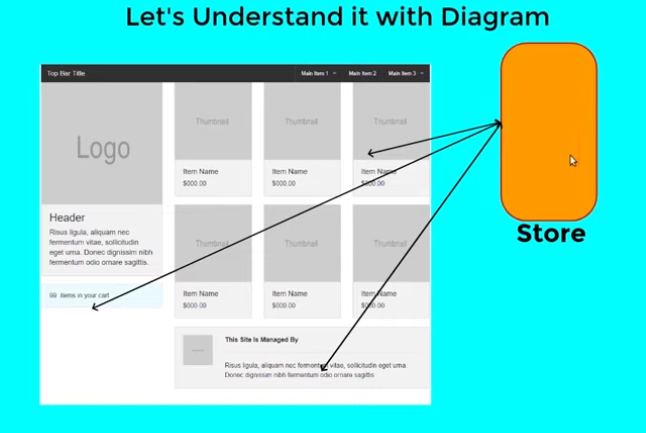


**State management tool – Application State**

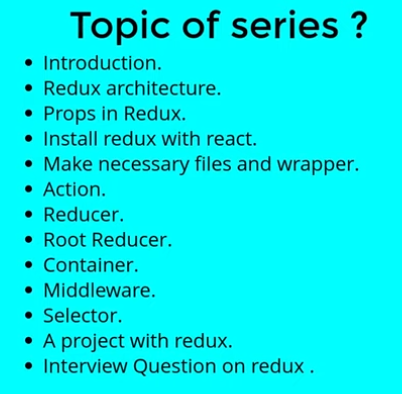
Why we need redux

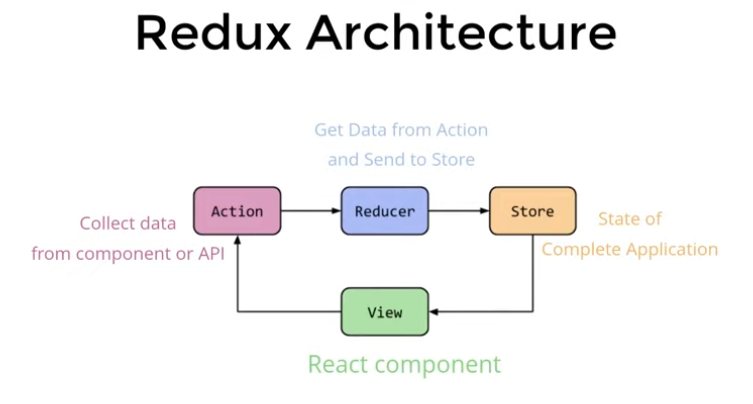
When we breaks component, we need to makes communication b/t components, so we need redux,

Using props drilling



Howto create store- so we need redux wrapper for whole application





Login using API Firebase

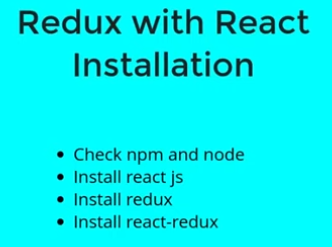
Reducer have two major work send data to store and

Reducer - filter the data as per store can receive- Filter means creation of new list of data or update the previous list, **so this information will come from Action using Type key**

**Store Can send data to any view (Component)**

**Each Application only have one store**

**Data unidirection only flow – request , action, reducer, store to view/component**

****

**$ npx create-react-app reaactRedux**

**Cd rreduxapp**

**Npm start**

$ npm i redux

$ npm i react-redux

3 library installed

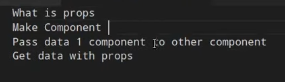
1. React
2. Redux
3. React-redux- communication channel for react and redux

Note: when we takes data into redux, comes through props and use through into view using props

**React Thumb rule-**

When we pass data into same component we use state

When we pass data into different component we use props



**App.js**

import './App.css';

import User from './components/User';

function App() {

const user={

  name:"manoj",

  age:38

}

  return (

    <div className="App">

         <h3>App Component</h3>

         <User data={user}/>

    </div>

  );

}

export default App;

**component/User.js**

import React from 'react'

const User = (props) => {

    //oneway console.warn(props);

    //secondway const {data}=props;

  return (

    <div>

      //oneway <h3>User Component {props.data.name}</h3>

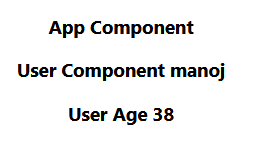
      //secondway <h3>User Component {data.name}</h3>

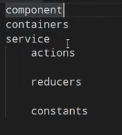
    </div>

  )

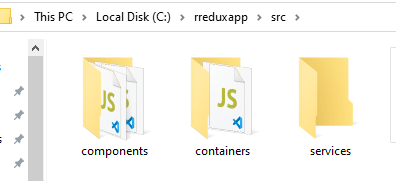
}

export default User





1. **Components – all components like header, footer, sidebar**
2. **Containers – will connect react & Reudx means components => Service**
3. **Service**
   1. **Actions**
   2. **Reducers**
   3. **Constants – will tell action and reducer file, which data will go to which action**

****

**Components/Home.js**

**Containers/HomeContainer.js**

****

****

**So after structure creation create first component Home.js**

import React from 'react'

const Home = () => {

  return (

    <div>

      <h1>home</h1>

    </div>

  )

}

export default Home

**App.js**

import './App.css';

import Home from './components/Home';

function App() {

  return (

    <div className="App">

         <h3>App Component</h3>

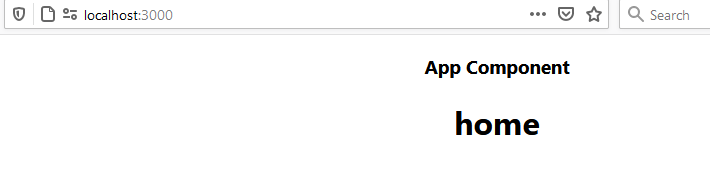
         <Home/>

    </div>

  );

}

export default App;

****

**Generally we do like this but now we will use Home.js component using containers/HomeContainer.js and then use into App.js for connectivity of Redux**

**HomeContainer.js**

import Home from '../components/Home';

export default Home;

**now we will call HomeContainer into App.js**

**App.js**

import './App.css';

import HomeContainer from './containers/HomeContainer';

function App() {

  return (

    <div className="App">

         <h3>App Component</h3>

        <HomeContainer/>

    </div>

  );

}export default App;

Now from now we will create each components like this use through containers not directly to components in App.js

**Components/Home.js**

import React from 'react'

function Home(props) {

    return (

        <div>

           <div className="add-to-cart">

    <span className="cart-count"></span>

                <img src="https://static.vecteezy.com/system/resources/thumbnails/000/496/007/small/Ecommerce\_998.jpg" />

            </div>

            <div className="cart-wrapper">

                <div className="img-wrapper item">

                    <img src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTkGFHaxpgvaIvB0ncj9d4TSHYMpKqKFQ0ItA&usqp=CAU" />

                </div>

                <div className="text-wrapper item">

                    <span>

                        I-Phone

                    </span>

                    <span>

                        Price: $1000.00

                    </span>

                </div>

                <div className="btn-wrapper item">

                    <button>Add To Cart</button>

                </div>

            </div>

            <div className="cart-wrapper">

                <div className="img-wrapper item">

                    <img src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTkGFHaxpgvaIvB0ncj9d4TSHYMpKqKFQ0ItA&usqp=CAU" />

                </div>

                <div className="text-wrapper item">

                    <span>

                        Samsung

                    </span>

                    <span>

                        Price: $1000.00

                    </span>

                </div><div className="btn-wrapper item">

                    <button>Add To Cart</button>

                </div>

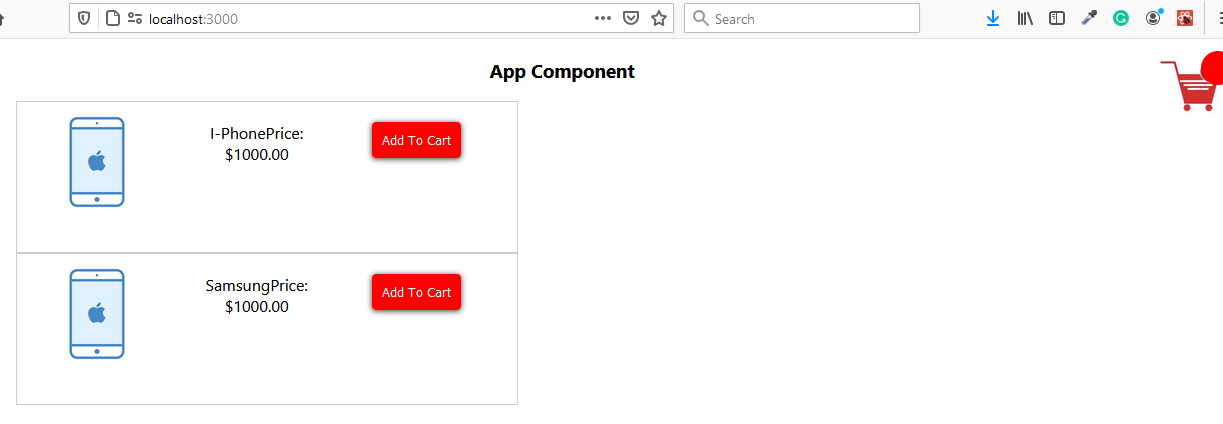
            </div>

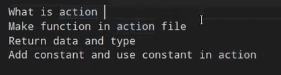
        </div>

    )

}

export default Home

****

****

**Action will send data from component to Store**

**View will click action and need to tell actions/actions.js which functionality we are making**

**And that function we will send some data and return with two values**

1. **Data:data**
2. **Type: typename**

export const addToCart=(data)=>

{

    return {

        data:data,

        type:'ADD\_TO\_CART'

    }

}

export const removeToCart=(data)=>

{

    return {

        data:data,

        type:'REMOVE\_TO\_CART'

    }

}

Type value should be string and uppercase

**View->action->reducer**

**But we will goto constants will same constants name will be in action and reducer**

**C:\rreduxapp\src\services\constants.js**

export const ADD\_TO\_CART="ADD\_TO\_CART";

export const REMOVE\_TO\_CART="REMOVE\_TO\_CART";

**now these constants will get into action file**

**C:\rreduxapp\src\services\actions\actions.js**

import {ADD\_TO\_CART,REMOVE\_TO\_CART} from '../constants';

export const addToCart=(data)=>

{

    return {

        data:data,

        type: ADD\_TO\_CART

    }

}

export const removeToCart=(data)=>

{

    return {

        data:data,

        type: REMOVE\_TO\_CART

    }

}

**Now view-action complete, next process is reducer**

**Reducer will take data from action and send and push to Store**

**Reducer will filter the data in push to store**

**Constants.js same file will use in actions.js and reducer.js for get constant name**

**We have two files in this path**

**C:\rreduxapp\src\services\reducers**

1. **Index.js- means root reducer**
2. **Reducer.js ( we can create any action reducer file) like addToCart**

**Reducer working**

1. Set initial of state and for store
2. Type of values- array or object ?
3. Switch cases based on type

export default function cartItems()

{

    return '';

}

**In this function we will pass two values**

1. **Initial state of data, should be array**
2. **ActionType we will pass in switch case**

const intialState={

    cartData:[]

}

export default function cartItems(intialState,action)

{

}

**Updated verison**

import { ADD\_TO\_CART, REMOVE\_TO\_CART } from '../constants';

const intialState = {

    cartData: []

}

export default function cartItems(state = intialState, action) {

    switch (action.type) {

        case ADD\_TO\_CART:

            return {

                ...state,

                cartData: action.data

            }

        case REMOVE\_TO\_CART:

            return {

                ...state,

                cartData: action.data

            }

            break;

        default: return state;

    }

}

**All reducer should be merge into rootreducer means C:\rreduxapp\src\services\reducers index.js**

**In root reducer we will use combinedReducers from redux**

**Index.js**

import {combineReducers} from 'redux';

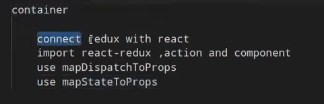
import cartItems from './reducer'; //function name in reducer

export default combineReducers({

    cartItems

});

**Now we have completed React & redux b/t communication is not build, so we need it, for this we will container file, which will be responsible for communication b/t react & redux**

****

**Because we have to push values into store and show in view**

**Connect function will have two argument**

1. **mapStateToProps – when we send data from Store to component**
2. **mapDispatchToProps- when we send data from component to store**
3. **Component name**

**C:\rreduxapp\src\containers\HomeContainer.js**

import Home from '../components/Home';

export default Home;

**Now we will update code of this**

1. **Import {connect} from ‘react-redux’;**
2. **Import component which we are working for action-** import Home from '../components/Home';
3. **Import action file** import {addToCart} from '../services/actions/actions';
4. **mapDispatchToProps function use**
5. **mapStateToProps function use**
6. **use of connect function with component name**

import Home from '../components/Home';

import {connect} from 'react-redux';

import {addToCart} from '../services/actions/actions';

const mapDispatchToPros=dispatch=>({

  //this event name will be use in view as event name addToCartHandler, means our add to cart event will call action with using data which we will dispatch

  addToCartHandler:data=>dispatch(addToCart(data))

})

const mapStateToPros=state=>({

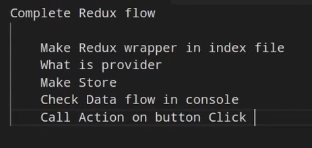
})

export default connect(mapStateToPros,mapDispatchToPros)(Home);

**Now our container communicating action, reducer with react**

Error: Could not find "store" in the context of "Connect(Home)". Either wrap the root component in a <Provider>, or pass a custom React context provider to <Provider> and the corresponding React context consumer to Connect(Home) in connect options.

**C:/rreduxapp/src/index.js:7**

****

**So index.js**

1. **import {createStore} from ‘redux’;**
2. **import {provider} from ‘react-redux’**
3. import rootReducer from './services/reducers/index';
4. const store=createStore(rootReducer);
5. <Provider store={store}><App/></Provider>

import React from 'react';

import ReactDOM from 'react-dom';

import './index.css';

import App from './App';

import reportWebVitals from './reportWebVitals';

//Start Redux work

import {createStore} from 'redux';

import {Provider} from 'react-redux';

import rootReducer from './services/reducers/index';

const store=createStore(rootReducer);

// End Redux work

ReactDOM.render(

  <Provider store={store}>

    <App />

  </Provider>,

  document.getElementById('root')

);

reportWebVitals();

**now final work is- create event and bind with addtocart button which is in Home.js component**

**How our Home.js component connected, it was connected using HomeContainer.js**

mapDispatchToPros will call automatically and

addToCartHandler:data=>**dispatch(addToCart(data))** will set which function will call with data, **so we will get data into Home.js using props**

import React from 'react'

function Home(props) {

  console.log(props);

    return (

        <div>

           <div className="add-to-cart">

    <span className="cart-count"></span>

                <img src="https://static.vecteezy.com/system/resources/thumbnails/000/496/007/small/Ecommerce\_998.jpg" />

            </div>

            <div className="cart-wrapper">

                <div className="img-wrapper item">

                    <img src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTkGFHaxpgvaIvB0ncj9d4TSHYMpKqKFQ0ItA&usqp=CAU" />

                </div>

                <div className="text-wrapper item">

                    <span>

                        I-Phone

                    </span>

                    <span>

                        Price: $1000.00

                    </span>

                </div>

                <div className="btn-wrapper item">

                    <button>Add To Cart</button>

                </div>

            </div>

            <div className="cart-wrapper">

                <div className="img-wrapper item">

                    <img src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTkGFHaxpgvaIvB0ncj9d4TSHYMpKqKFQ0ItA&usqp=CAU" />

                </div>

                <div className="text-wrapper item">

                    <span>

                        Samsung

                    </span>

                    <span>

                        Price: $1000.00

                    </span>

                </div>

                <div className="btn-wrapper item">

                    <button>Add To Cart</button>

                </div>

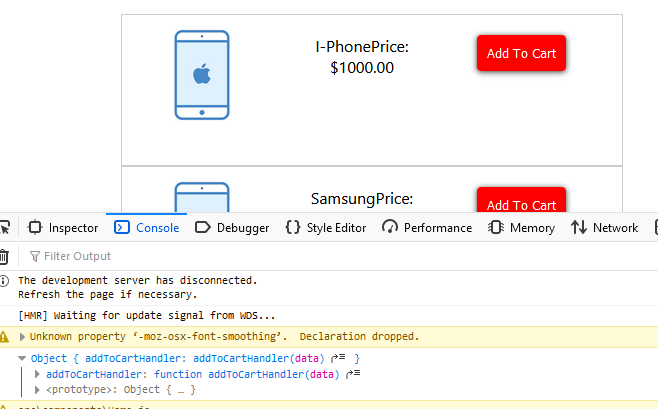
            </div>

        </div>

    )

}

export default Home

****

<button

                    onClick={()=>props.addToCartHandler({price:'1000', name:'I-Phone'})  }

                    >Add To Cart</button>

**Now run the code, but how we will check action is called or not, how we will get which action is called, check communication file means container file HomeContainer.js**

 addToCartHandler:data=>dispatch(addToCart(data))

**showing action addToCart is called- C:\rreduxapp\src\services\actions\actions.js**

export const addToCart=(data)=>

{

     console.log("action",data);

    return {

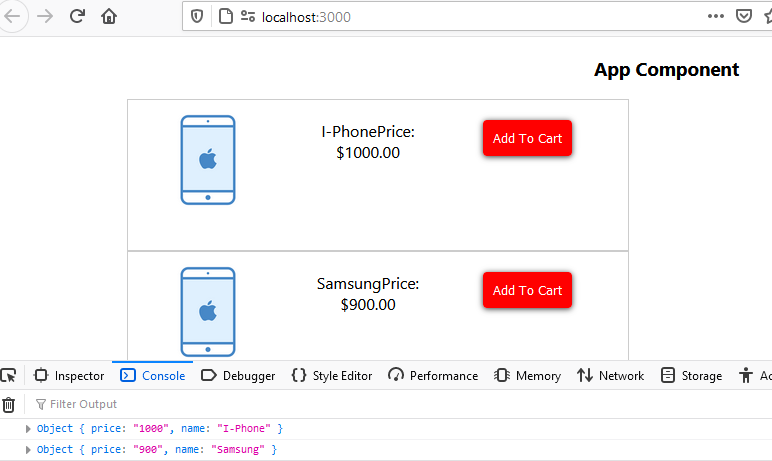
        data:data,

        type: ADD\_TO\_CART

    }

}

**After click on add to cart button action will fire**

****

**If we see data is coming into action, lets check this data is going to reducer or not**

**C:\rreduxapp\src\services\reducers\reducer.js**

import { ADD\_TO\_CART, REMOVE\_TO\_CART } from '../constants';

const intialState = {

    cartData: []

}

export default function cartItems(state = intialState, action) {

    switch (action.type) {

        case ADD\_TO\_CART:

**console.warn("reducer",action);**

            return {

                ...state,

                cartData: action.data

            }

        case REMOVE\_TO\_CART:

            return {

                ...state,

                cartData: action.data

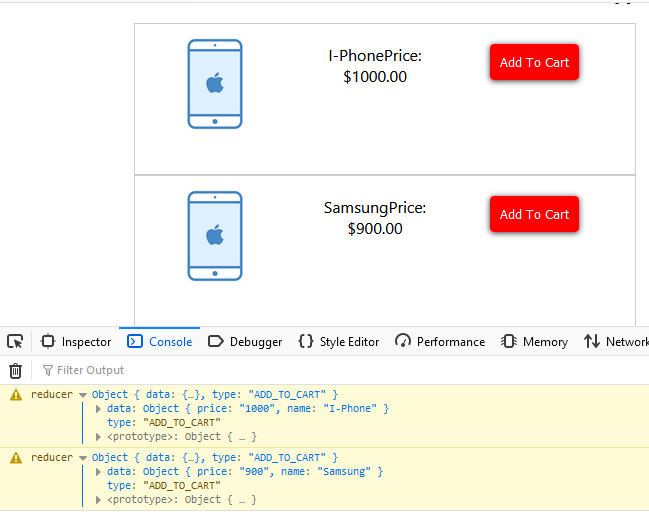
            }

            break;

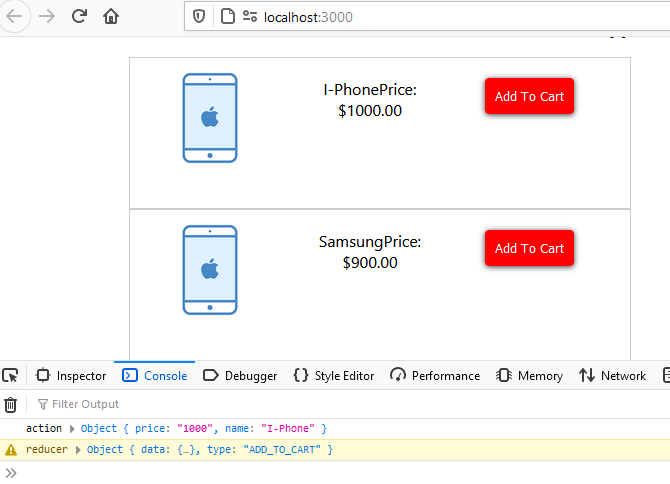
        default: return state;

    }

}

****

**Or**

****

**Now data should get on Cart Icon means mapStateToProps will run**

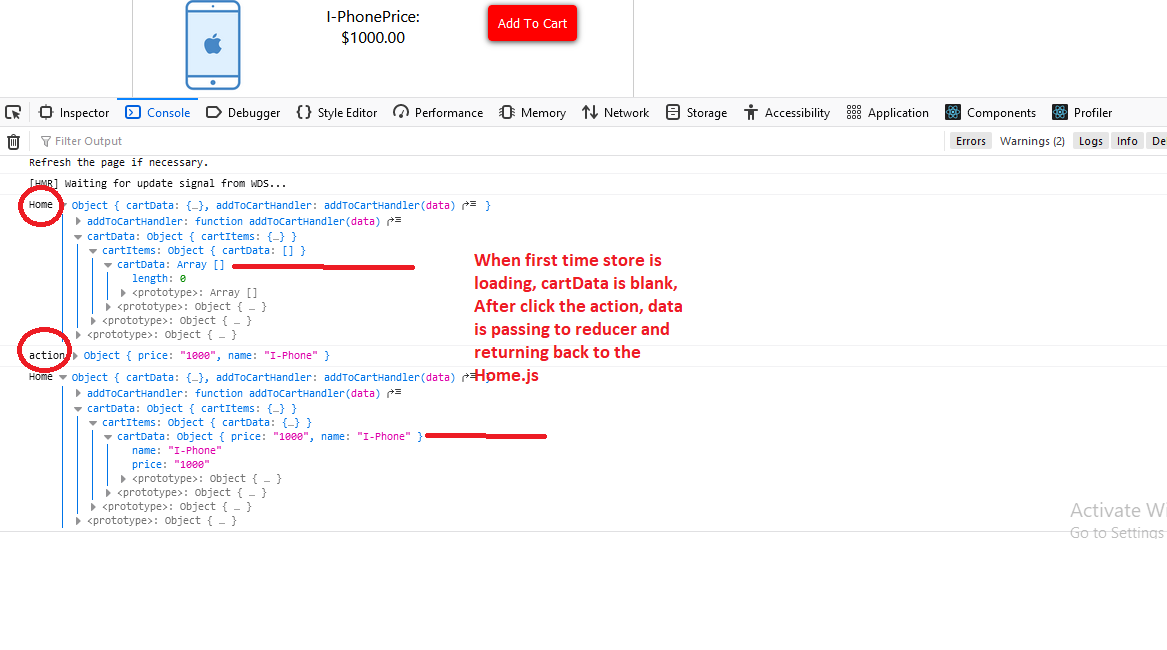
**C:\rreduxapp\src\containers\HomeContainer.js**

const mapStateToPros=state=>({

    cartData:state

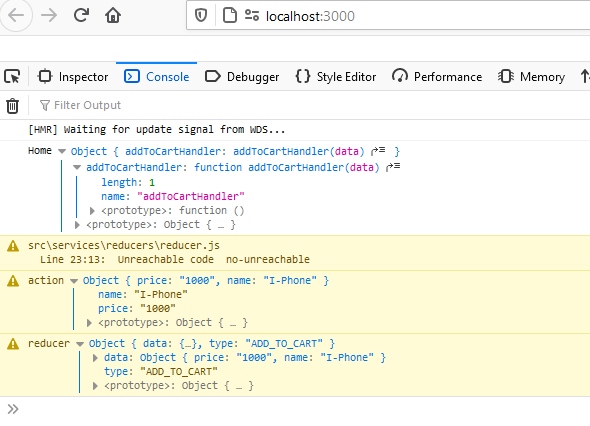
})

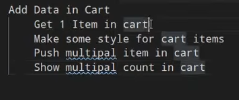
**Now we want show data in cart icon, which is in Home.js, in the form of props**

****

**First time store set blank and when we click on addToCartHandler mapDispatchToProps and call action and set values of state and send back to reducer, reducer receives data and set to store and**

**mapStateToProps run and send back to all components where we required we can get using props.**

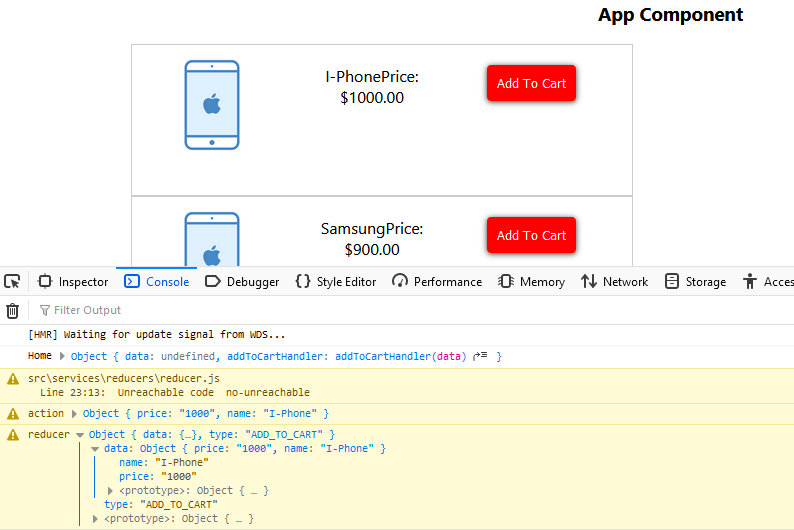
****

****

**How to minimize the logs to check store data and process**

**HomeContainer.js**

**Update code**

****

**HomeContainer.js**

//return data object which will set current state of rootreducer name, where cartData is object return after reducer action perform

const mapStateToPros=state=>({

    data:state.cartItems

})

In Home.js

import React from 'react'

function Home(props) {

**console.warn("Home",props.data);**

    return (

        <div>

           <div className="add-to-cart">

    <span className="cart-count"></span>

                <img src="https://static.vecteezy.com/system/resources/thumbnails/000/496/007/small/Ecommerce\_998.jpg" alt="" />

            </div>

            <div className="cart-wrapper">

                <div className="img-wrapper item">

                    <img src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTkGFHaxpgvaIvB0ncj9d4TSHYMpKqKFQ0ItA&usqp=CAU" alt="" />

                </div>

                <div className="text-wrapper item">

                    <span>

                        I-Phone

                    </span>

                    <span>

                        Price: $1000.00

                    </span>

                </div>

                <div className="btn-wrapper item">

                    <button

                    onClick={()=>props.addToCartHandler({price:'1000', name:'I-Phone'})  }

                    >Add To Cart</button>

                </div>

            </div>

            <div className="cart-wrapper">

                <div className="img-wrapper item">

                    <img src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTkGFHaxpgvaIvB0ncj9d4TSHYMpKqKFQ0ItA&usqp=CAU" alt="" />

                </div>

                <div className="text-wrapper item">

                    <span>

                        Samsung

                    </span>

                    <span>

                        Price: $900.00

                    </span>

                </div>

                <div className="btn-wrapper item">

                    <button  onClick={()=>props.addToCartHandler({price:'900', name:'Samsung'})  }>Add To Cart</button>

                </div>

            </div>

        </div>

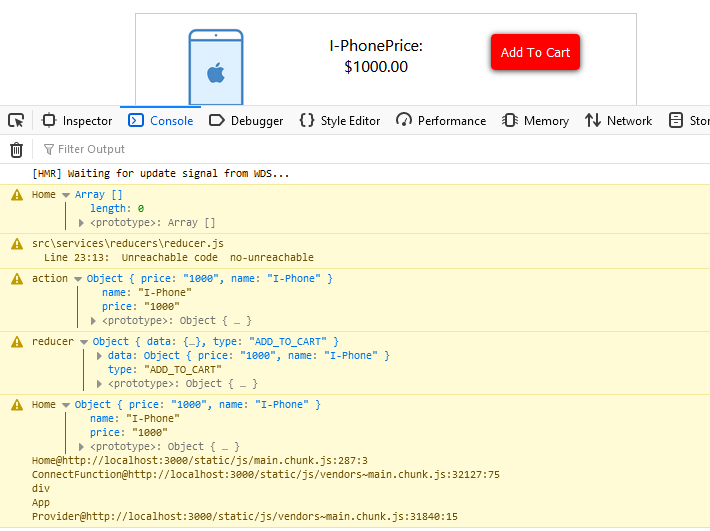
    )

}

export default Home

**Means in data object we are getting item name and price and type means action nameif we want sto remove this data object like above image**

**Goto Home.js**

****

**Now we are getting only single object values but cart items can be multiple**

**Reducer.js**

import { ADD\_TO\_CART, REMOVE\_TO\_CART } from '../constants';

const intialState = {

    cartData: []

}

export default function cartItems(state = [], action) {

    switch (action.type) {

        case ADD\_TO\_CART:

        console.warn("reducer",action);

            return **[**

                ...state,

**{ cartData: action.data}**

**]**

        case REMOVE\_TO\_CART:

            return {

                ...state,

                cartData: action.data

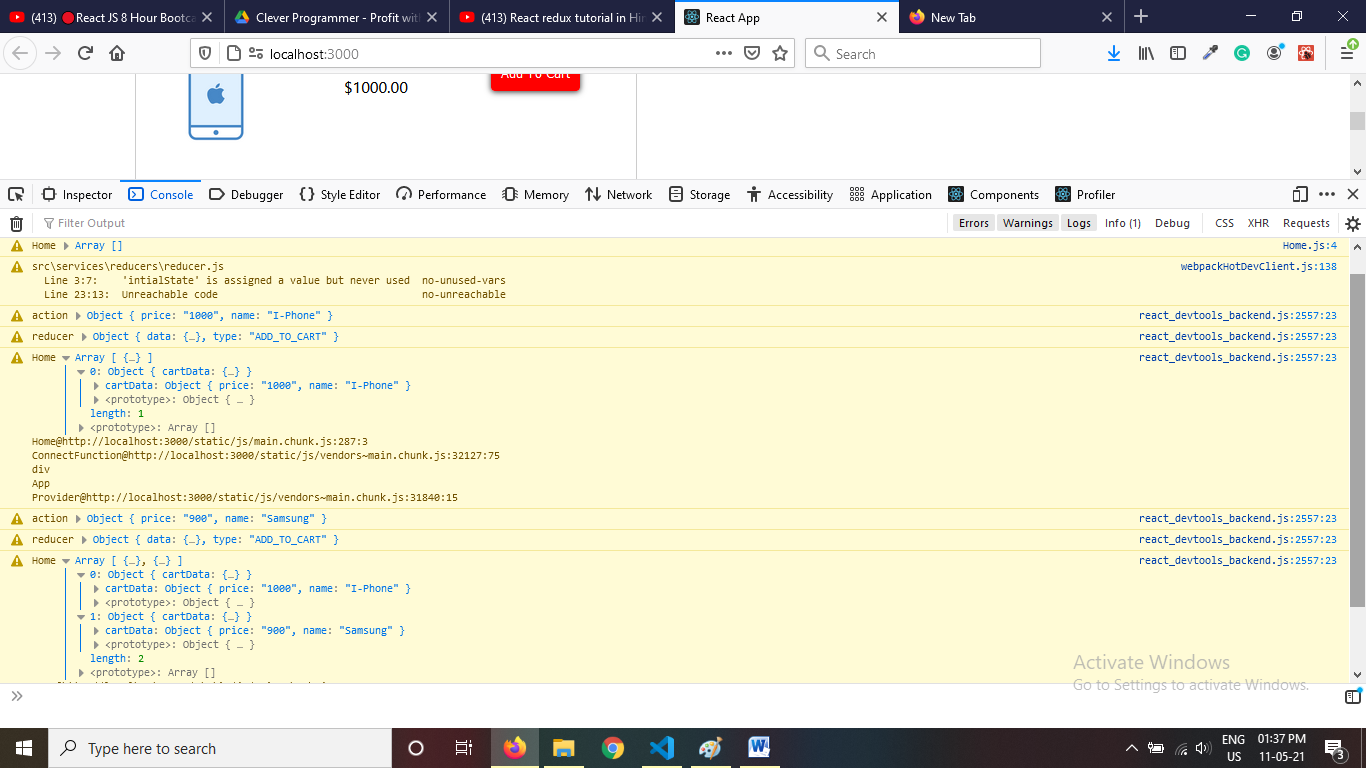
            }

            break;

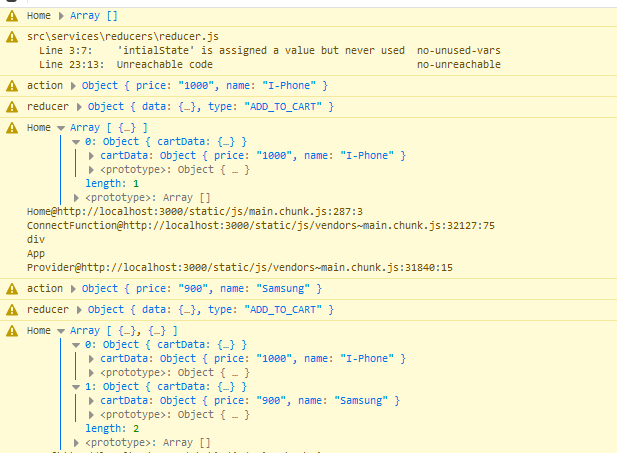
        default: return state;

    }

}



**Now we are getting array when we click both items to be add in cart**

****

**Home.js**

import React from 'react'

function Home(props) {

  console.warn("Home",props.data);

    return (

        <div>

           <div className="add-to-cart">

**<span className="cart-count">{props.data.length}</span>**

                <img src="https://static.vecteezy.com/system/resources/thumbnails/000/496/007/small/Ecommerce\_998.jpg" alt="" />

            </div>

            <div className="cart-wrapper">

                <div className="img-wrapper item">

                    <img src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTkGFHaxpgvaIvB0ncj9d4TSHYMpKqKFQ0ItA&usqp=CAU" alt="" />

                </div>

                <div className="text-wrapper item">

                    <span>

                        I-Phone

                    </span>

                    <span>

                        Price: $1000.00

                    </span>

                </div>

                <div className="btn-wrapper item">

                    <button

                    onClick={()=>props.addToCartHandler({price:'1000', name:'I-Phone'})  }

                    >Add To Cart</button>

                </div>

            </div>

            <div className="cart-wrapper">

                <div className="img-wrapper item">

                    <img src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTkGFHaxpgvaIvB0ncj9d4TSHYMpKqKFQ0ItA&usqp=CAU" alt="" />

                </div>

                <div className="text-wrapper item">

                    <span>

                        Samsung

                    </span>

                    <span>

                        Price: $900.00

                    </span>

                </div>

                <div className="btn-wrapper item">

                    <button  onClick={()=>props.addToCartHandler({price:'900', name:'Samsung'})  }>Add To Cart</button>

                </div>

            </div>

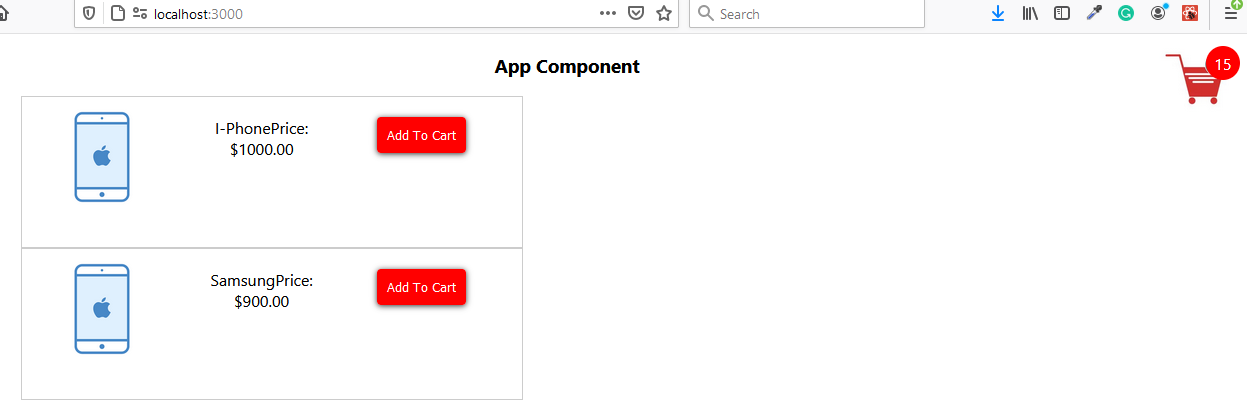
        </div>

    )

}

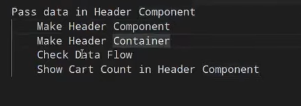
export default Home

**Above code we are able to count cart items**

****

**Now we have only one component Home.js**

**But we will break into two so we can get idea how we can get same state into other component using redux**

****

**For this break Home.js cut code and make new Header.js component**

<div className="add-to-cart">

    <span className="cart-count">{props.data.length}</span>

                <img src="https://static.vecteezy.com/system/resources/thumbnails/000/496/007/small/Ecommerce\_998.jpg" alt="" />

            </div>

**Header.js**

import React from 'react'

const Header = (props) => {

    console.log("Header component",props);

  return (

    <div>

    <div className="add-to-cart">

    <span className="cart-count">{props.data.length}</span>

                <img src="https://static.vecteezy.com/system/resources/thumbnails/000/496/007/small/Ecommerce\_998.jpg" alt="" />

            </div>

    </div>

  )

}

export default Header

Now cart icon and count will not show into home.js or view page, for this we have to make container which will access of this components/Header.js

**C:\rreduxapp\src\containers\HeaderContainer.js**

import Header from '../components/Header';

import { connect } from 'react-redux';

//in this we just getting data not sending the data so only mapDispatchToProps will be blank

const mapDispatchToPros = dispatch => ({

})

//return data object which will set current state of rootreducer name, where cartData is object return after reducer action perform

const mapStateToPros = state => ({

  data: state.cartItems

})

export default connect(mapStateToPros, mapDispatchToPros)(Header);

Now we have to show our header component into app, App.js

import './App.css';

import HomeContainer from './containers/HomeContainer';

import HeaderContainer from './containers/HeaderContainer';

function App() {

  return (

    <div className="App">

         <h3>App Component</h3>

         <HeaderContainer/>

        <HomeContainer/>

    </div>

  );

}

export default App;

according like this we can pass data from store to any component.

**Remove items functionality**

<button onClick={() => props.removeToCartHandler()}>Remove Cart</button>

**Actions.js**

import { ADD\_TO\_CART, REMOVE\_TO\_CART } from '../constants';

export const addToCart = (data) => {

    console.warn("Add to Cart Action", data);

    return {

        data: data,

        type: ADD\_TO\_CART

    }

}

export const removeToCart = (data) => {

    console.warn("Remove from Cart Action", data);

    return {

        data: data,

        type: REMOVE\_TO\_CART

    }

}

**Constants.js**

export const ADD\_TO\_CART="ADD\_TO\_CART";

export const REMOVE\_TO\_CART="REMOVE\_TO\_CART";

**C:\rreduxapp\src\containers\HomeContainer.js**

import {addToCart,removeToCart} from '../services/actions/actions';

const mapDispatchToPros=dispatch=>({

  //this event name will be use in view as event name addToCartHandler, means our add to cart event will call action with using data which we will dispatch

    addToCartHandler:data=>dispatch(addToCart(data)),

**removeToCartHandler:data=>dispatch(removeToCart(data))**

})

Now check in Home.js we are able to get action in store

Now we have change function of remove item, that is in reducer.js

import { ADD\_TO\_CART, REMOVE\_TO\_CART } from '../constants';

const intialState = {

    cartData: []

}

export default function cartItems(state = [], action) {

    switch (action.type) {

        case ADD\_TO\_CART:

        console.warn("reducer",action);

            return [

                ...state,

               { cartData: action.data}

            ]

        case REMOVE\_TO\_CART:

            state.pop();

            return [

                ...state

            ]

            break;

        default: return state;

    }

}

**Now we are able to add and remove, but in remove we are not getting which one we are deleting or adding, just showing count of store array**