



Redux Example



Sudhakar Sharma • Jul 27

MVC

- Model -View - Controller

Model

- It represents the data we are working with.
- It contains data logic.
- It contains data validation rules
- It contains meta data [changes and updates]
- It is referred as "single-source-of-truth"

View

- UI
- It defines view and partial view
- It triggers the actions

Controller

- It comprises actions to perform
- It is referred as subscriber
- It is a reducer.
- It handles the overall communication between model and view.

User Clicks on "Add to Cart" => Collect the Product Details => Store in Redux Store => Assign to CartItems[] => Bind to UI

Step-1: Install Redux Tool Kit for your project

```
> npm install @reduxjs/toolkit react-redux --save
```

Install redux dev tools for your chrome browser

Step-2: Create a Redux slicer

- In redux slicer is used to configure the initial state and actions.

```
cartItems:any[] = []  
cartCount:number=0  
addToCart:any;  
createSlicer
```

- Add a new file into project



"CartSlicer.tsx"



```
const initialState = {
  cartItems: [],
  cartCount : 0
}

const cartSlice = createSlice({
  name: 'cart',
  initialState,
  reducers: {
    addToCart(state:any, action){
      state.cartItems.push(action.payload);
      state.cartCount = state.cartItems.length;
    }
  }
});
export const { addToCart } = cartSlice.actions;
export default cartSlice.reducer;
```

Note: Slice comprises of

- a) initial state
- b) name
- c) reducer : actions

Step-3: Configure Store

"store.tsx"

```
import { configureStore } from "@reduxjs/toolkit";
import CartSlicer from "../CartSlicer";

export default configureStore({
  reducer: {
    store : CartSlicer //initial state[cartItems, count], actions
  }
})
```

Note: It creates a store centralized and adds the references for

- a) cartItems[]
- b) carCount

Step-4: Configure UI

"ProductsComponent.tsx" [shopping cart page] with Add to cart

```
import { addToCart } from "../CartSlicer";
import { useDispatch } from "react-redux";
```

```
const dispatch = useDispatch();
```



```

{
  products.map(product=>
    <card>
      <Button onClick={()=> handleAddToCart(product)} className="w-100" variant="contained" >
<span className="bi bi-cart4"></span> Add to Cart</Button>

    </card>
  )
}

```

Step-5: Configure store at app level with a provider

index.tsx

```

import { Provider } from 'react-redux';
import store from './ishop/store'; [store.tsx]

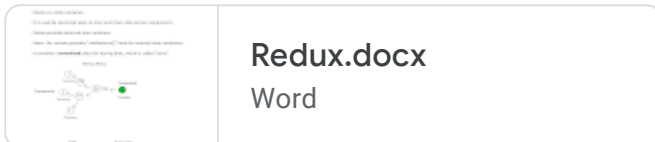
```

```

<React.StrictMode>
  <Provider store={store} >
    <IshopIndexComponent />
  </Provider>
</React.StrictMode>,

```

React Native, Ionic - Mobile - PWA



Redux.docx
Word



Class comments

