How to use Context vs Redux In Reactjs.

READ NOW →

@VEDANG NIKURE

React Context

1. Context is the Internal core part of the React, we don't need to install external library for it.

2. Create a context using the createContext()

```jsx
import { createContext } from "react";

const UserContext = createContext({
  loggedInUser: "Default User",
});

export default UserContext;
```

3. Then use that particular context uisng useContext()

```jsx
import React from "react";
import { useContext } from "react";
import UserContext from "../utils/UserContext";

const UsingContext = () => {
  const { loggedInUser } = useContext(UserContext);
  return <div>{loggedInUser}</div>;
};

export default UsingContext;
```

4.After that provide the context to whole App/ Particular portion by Wrapping with Provider.

```jsx
import React from "react";
import UserContext from "../utils/UserContext";
import UserContext from "./utils/UserContext";
const UsingContext = () => {
  const [userName, setUserName] = useState();
  return (
    <UserContext.Provider value={{ loggedInUser: userName, setUserName }}>
      <YourComponentsHere />
    </UserContext.Provider>
  );
};

export default UsingContext;
```

Redux

1. Redux is not an internal part of React, we have to install 2 library for it.

- React Redux :- It is like bridge between React and Redux
- Redux Toolkit :- It is core redux Library

2. Need to create the slice using the createSlice({...}) in that we need to provide the “name”, “initialState”, “reducers” and in the reducers we need to pass the actions.
In the actions we have state and action.

```
import { createSlice } from "@reduxjs/toolkit";
const cartSlice = createSlice({
  name: "cart",
  initialState: {
    items: [],
  },
  reducers: {
    addItem: (state, action) => {
      state.items.push(action.payload);
    },
    removeItem: (state, action) => {
      state.items.pop();
    },
    clearCart: (state, action) => {
      return { items: [] };
    },
  },
});

export const { addItem, removeItem, clearCart } = cartSlice.actions;
export default cartSlice.reducer;
```

3. Need to configure the store using the configureStore(), In that we need to pass the sline in the reducer

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

const appStore = configureStore({
  reducer: {
    cart: cartReducer,
  },
});

export default appStore;
```

4. To provide the redux, we need to wrap to whole App/ Particular portion Wrapping with Provider and we have to provide the configureStore name as prop in the store

```jsx
import React from "react";
import { Provider } from "react-redux";
import appStore from "./utils/appStore";
const APP = () => {
  return (
    <Provider store={appStore}>
      <YourComponentsHere />
    </Provider>
  );
};

export default APP;
```

5. To select the specific portion of the store use useSelector() Hook

```jsx
import { useSelector } from "react-redux";
const Cart = () => {
  // const cartItems = useSelector((store) => store.Name_of_slice.InitialValue_of_slice);
  const cartItems = useSelector((store) => store.cart.items);
  return <div>{cartItems}</div>;
};

export default Cart;
```

6. To perform/ Dispatch an action use the hook

useDispatch()

```
import { clearCart } from "../utils/cartSlice";
import { useDispatch } from "react-redux";
const Cart = () => {
  const dispatch = useDispatch();

  const handleClearCart = () => {
    dispatch(clearCart());
  };
  return (
    <button
      className=" p-2 m-2 bg-black text-white rounded-lg"
      onClick={handleClearCart}>
      Clear Cart
    </button>
  );
};

export default Cart;
```