**Redux New Version : Reduc-Tool-Kit ( RTK )**

1. **Installation :**

|  |
| --- |
| npm install @reduxjs/toolkit  npm install react-redux |

1. **Create Slice :**

|  |
| --- |
| import { createSlice } from "@reduxjs/toolkit";  const counterSlice = createSlice({      name: "counter-slice",      initialState: 0,      reducers: {          incrementCounter(state, action) {              return state + action.payload          },          decrementCounter(state, action) {              return state - 1          },          clearCounter(state, action) {              return state \* 0          }      }  })  // console.log(counterSlice.actions); 🡨 this will return all the action creators with its types.  export default counterSlice;  export const { incrementCounter, decrementCounter, clearCounter } = counterSlice.actions; |

1. **Create Store :**

|  |
| --- |
| import { configureStore } from "@reduxjs/toolkit";  **// Importing Reducers From CounterSlice**  import counterSlice from "./slices/counterSlice";  const store = configureStore({  **// Use This When ther is only one slice.**      // reducer: counterSlice.reducer,  **/\***  **Use This When there are more then one slice.**  **reducer : {**  **reducerName : counterSlice.reducer, // reducerName is a micro reducer.**  **}**  **\*/**      reducer: {          micro\_reducer\_counter: counterSlice.reducer,      },  })  export default store; |

1. **Connect Redux With React :**

|  |
| --- |
| import React from 'react'  import { useDispatch, useSelector } from 'react-redux';  import { clearCounter, decrementCounter, incrementCounter } from "./redux/slices/counterSlice";  export default function App() {  **const dispatch = useDispatch();**  **const myCounter = useSelector((data) => data.micro\_reducer\_counter); 🡨 used to display data from store**    return (      <div>        <h1>Counter : **{myCounter}</**h1>        <button type="button**" onClick={() => dispatch(decrementCounter())}**>Decrement --</button>        &nbsp;        <button type="button" **onClick={() => dispatch(incrementCounter(100))}** >Increment ++</button>        &nbsp;        <button type="button" **onClick={() => dispatch(clearCounter())}** >Clear Counter</button>      </div>    )  } |