**Redux and Redux Toolkit – Documents**

Redux Toolkit Configurations :-

­­Step 1):- Configure store – we need to configure the store which holds all the reducers and reducers are the function which updates the store state based on the action dispatched

import {configureStore} from '@reduxjs/toolkit'

export const store = configureStore({

    reducer:{}

})

Step 2) : Wrap the whole App with Provider which will inject this whole store into the each and every component of the application

import { store } from './store/store.js';

import { Provider } from 'react-redux';

ReactDOM.createRoot(document.getElementById('root')).render(

  <React.StrictMode>

    <Provider store={store}>

      <App />

    </Provider>

  </React.StrictMode>

);

Step 3: We need to create multiple slice . A Slice can be considered to the different reducers which has will listen to particular type of the action and update the stores.

import { createSlice } from "@reduxjs/toolkit";

const counterInitialState={

    count:0

}

const counterSlice = createSlice({

    name:'counter',

    initialState:counterInitialState,

    reducers:{

        increment:(state)=>{

            state.count++

        },

        decrement:(state)=>{

            state.count--;

        },

        reset:(state)=>{

            state.count=0

        }

    }

})

export const counterSliceActions = counterSlice.actions;

export const counterSliceReducer = counterSlice.reducer

Step 4: We need to combine multiple slice into one root reducers which is then passed to the store provider.

import { combineReducers } from "@reduxjs/toolkit";

import { counterSliceReducer } from "./slices/counterSlice";

export const rootReducer = combineReducers({

    counter:counterSliceReducer

})

Step 5: Now add the this root reducer to the main config store reducer

import { configureStore } from "@reduxjs/toolkit";

import rootReducer from "./rootReducer";

*// Store is the source truth for the application*

const store = configureStore({

    reducer:rootReducer

})

export default store

Step 6 : Now go to component import useSelector , useDispatch and actions of the slice create

import { useSelector, useDispatch } from 'react-redux';

*// local imports*

import { counterSliceAction } from '../slices/slice';

import styles from './Counter.module.css';

const CounterRedux = () => {

  const count = useSelector((store) => store.counter.count);

  const dispatch = useDispatch();

*//   handler functions*

  const onDecrementHandler = (event) => {

    if (count <= 0) return;

    dispatch(counterSliceAction?.decrement())

  };

  const onIncrementHandler = (event) => {

    dispatch(counterSliceAction?.increment());

  };

  const onResetHandler = (event) => {

    dispatch(counterSliceAction?.reset())

  };

  return (

    <div className={styles.counter\_container}>

      <h1>Counter Machine Codings</h1>

      <h1>{count}</h1>

      <div className={styles.button\_container}>

        <button onClick={onDecrementHandler}>Decrement</button>

        <button onClick={onIncrementHandler}>Increment</button>

        <button onClick={onResetHandler}>Reset</button>

      </div>

    </div>

  );

};

export default CounterRedux;

* useSelector – The useSelector hook provides us the part of the state from the store which we are interested in ,
* const count = useSelector((store) => store.counter.count);
* const dispatch = useDispatch();

In the above example from the store -> goes to counter slice -> From counter slice it gets the count state variable

* useDispatch – useDisptach helps us to create the dispatch object which helps to call the actions