App.js

import React, { Component } from 'react';

import './App.css';

import store, { COUNTER\_INCREMENT, COUNTER\_DECREMENT } from './Store';

class App extends Component {

  constructor(){

    super();

    this.state ={

      name : 'React'

    }

  }

  increment = () =>{

    store.dispatch({type:COUNTER\_INCREMENT});

    console.log(store.getState().count);

  }

  decrement = () =>{

    store.dispatch({type:COUNTER\_DECREMENT});

    console.log(store.getState().count);

  }

render(){

  return (

    <div >

      <h1>Hello {this.state.name}</h1>

      <button onClick={this.increment}> increment</button>

      <button onClick={this.decrement}> decrement</button>

    </div>

  );

}

}

export default App;

Store.js

import \* as redux from 'redux';

//Actions

export const COUNTER\_INCREMENT = 'counter/increment';

export const COUNTER\_DECREMENT = 'counter/decrement';

const initialState = {

    count : 0

}

const reducer = (state = initialState, action) => {

    switch(action.type){

        case COUNTER\_INCREMENT : {

            return { ...state, count: state.count + 1};

            // break;

        }

        case COUNTER\_DECREMENT : {

            return { ...state, count: state.count - 1};

            // break;

        }

        default : {

            return state;

        }

    }

}

const store =redux.createStore(reducer);

export default store;

More then Reducer:

App.js

import React, { Component } from 'react';

import './App.css';

import store, { COUNTER\_INCREMENT, COUNTER\_DECREMENT } from './Store';

class App extends Component {

  constructor(){

    super();

    this.state ={

      name : 'React'

    }

  }

  increment = () =>{

    store.dispatch({type:COUNTER\_INCREMENT});

    console.log(store.getState().count);

  }

  decrement = () =>{

    store.dispatch({type:COUNTER\_DECREMENT});

    console.log(store.getState().count);

  }

  push = () =>{

    store.dispatch({type: 'push', value : Math.random()});

    console.log(store.getState());

  }

  pop = () =>{

    store.dispatch({type:'pop'});

    console.log(store.getState());

  }

render(){

  return (

    <div >

            <h1>Hello {this.state.name}</h1>

      <button onClick={this.increment}> increment</button>

      <button onClick={this.decrement}> decrement</button>

      <button onClick={this.push}> Push</button>

      <button onClick={this.pop}> Pop</button>

    </div>

  );

}

}

export default App;

Store.js

import \* as redux from "redux";

export const COUNTER\_INCREMENT = "counter/increment";

export const COUNTER\_DECREMENT = "counter/decrement";

const countInitialState ={

    count : 0

}

const countReducer = (state = countInitialState, action) => {

    switch(action.type){

        case COUNTER\_INCREMENT : {

            return { ...state, count: state.count + 1};

            // break;

        }

        case COUNTER\_DECREMENT : {

            return { ...state, count: state.count - 1};

            // break;

        }

        default : {

            return state;

        }

    }

}

const testReducer = (state = [], action) => {

    switch(action.type){

        case 'push' : {

            const data = [...state];

            data.push(action.value);

            return [...data];

             break;

        }

        case 'pop' : {

            const data = [...state];

            data.pop();

            return [...data];

             break;

        }

        default : {

            return state;

        }

    }

}

const rootReducer =(state = {} , action) =>{

    return {

        counter : countReducer(state.counter, action),

        tester : testReducer(state.tester, action)

    }

}

const store =redux.createStore(rootReducer);

export default store;