|  |  |  |
| --- | --- | --- |
| **actionTypes.js**  export const INCREMENT = 'INCREMENT';  export const DECREMENT = 'DECREMENT';  export const INCREMENTBYAMOUNT = 'INCREMENTBYAMOUNT';  export const NUMBERINFOLOADED = 'NUMBERINFOLOADED';  export const NUMBERINFOLOADFAILED = 'NUMBERINFOLOADFAILED'; | **Index.js**  import React from 'react';  import ReactDOM from 'react-dom';  import App from './App';  import { Provider } from 'react-redux';  import store from './store';  ReactDOM.render(  <Provider store={store}>  <React.StrictMode>  <App />  </React.StrictMode>  </Provider>,  document.getElementById('root')  ); | **Store.js**  import rootReducer from './reducers';  import { applyMiddleware, createStore, compose } from 'redux';  import ReduxThunk from 'redux-thunk';  const devtools =  window.\_\_REDUX\_DEVTOOLS\_EXTENSION\_\_ && window.\_\_REDUX\_DEVTOOLS\_EXTENSION\_\_();  const store = createStore(  rootReducer,  compose(applyMiddleware(ReduxThunk), devtools)  );  export default store; |
| **actions.js**  import \* as actions from './actionType';  import axios from 'axios';  export const increment = (count) => {  // fetch api  return {  type: actions.INCREMENT,  payload: count,  };  };  export const decrement = (count) => {  return {  type: actions.DECREMENT,  payload: count,  };  };  export const incrementByAmount = (count) => {  return {  type: actions.INCREMENTBYAMOUNT,  payload: count,  };  };  export const getNumberInfo = (num) => {  return async (dispatch) => {  const url = `http://numbersapi.com/${num}/trivia`;  const res = await axios.get(url);  const numberInfo = res.data;  try {  if (res.status === 200) {  dispatch({  type: actions.NUMBERINFOLOADED,  payload: numberInfo,  });  }  } catch (err) {  dispatch({  type: actions.NUMBERINFOLOADFAILED,  payload: err.message,  });  }  };  }; | **Reducers.js**  import \* as actions from './actionType';  import { combineReducers } from 'redux';  const initialCountState = {  count: 0,  };  const initalNumbersState = {  numberInfo: '',  error: '',  };  const countReducer = (state = initialCountState, action) => {  switch (action.type) {  case actions.INCREMENT: {  return {  ...state,  count: action.payload + 1,  };  }  case actions.DECREMENT: {  return {  ...state,  count: action.payload - 1,  };  }  case actions.INCREMENTBYAMOUNT: {  return {  ...state,  count: (state.count += action.payload),  };  }  default: {  return state;  }  }  };  const numberReducer = (state = initalNumbersState, action) => {  switch (action.type) {  case actions.NUMBERINFOLOADED: {  return {  ...state,  numberInfo: action.payload,  };  }  case actions.NUMBERINFOLOADFAILED: {  return {  ...state,  error: action.payload,  };  }  default: {  return state;  }  }  };  const rootReducer = combineReducers({  countReducer,  numberReducer,  });  export default rootReducer; | App.js  import React, { useState } from 'react';  import './App.css';  import { useDispatch, useSelector, useStore } from 'react-redux';  import {  decrement,  increment,  incrementByAmount,  getNumberInfo,  } from './actions';  const App = () => {  const store = useStore();  const dispatch = useDispatch();  // state values  const count = useSelector((state) => state.countReducer.count);  const numberInfo = useSelector((state) => state.numberReducer.numberInfo);  const error = useSelector((state) => state.numberReducer.error);  const [amount, setAmount] = useState(2);  console.log('store', store.getState());  return (  <div className="app">  <div className="counter">  <h1>Counter</h1>  <div>  <button onClick={() => dispatch(decrement(count))}> - </button>  <div className="count">{count} </div>  <button onClick={() => dispatch(increment(count))}> + </button>  </div>  </div>  <div>  <input  value={amount}  onChange={(e) => setAmount(Number(e.target.value) || 0)}  />  </div>  <div>  <button onClick={() => dispatch(incrementByAmount(amount))}>  {' '}  +{amount}{' '}  </button>  </div>  <div style={{ marginTop: '20px' }}>  <button onClick={() => dispatch(getNumberInfo(count))}>  Click to learn more about this number  </button>  </div>  <div style={{ marginTop: '20px' }}>{error ? error : numberInfo}</div>  </div>  );  };  export default App; |
|  |  |  |
|  |  |  |