Redux & Reactjs

CÁC BƯỚC TẠO Web DÙNg Redux

1. Cài ĐặT Thư VIỆN Redux, React-Redux

2. Tại File Index.Js

2.1. Import Creatstore Và Provider

```
import { createStore } from "redux";
import { Provider } from "react-redux";
```

2.2. TạO Rootreducer ChuÃ CÁC Reduser Con

TạO File Rootreducer Trong Thư MuC Reducers

```
redux
  action
  constants
  reducers
    indexReducer.js
    userReducer.js
```

Trong Rootreducer Import Combinereducers Để Gom CÁC Reduser Con Lại

```
import { combineReducers } from "redux";
import UserReducer from "./UserReducer";

const rootReducer = combineReducers({
  UserReducer //UserReducer: UserReducer
});

export default rootReducer;
```

3. Trở Về File Index.js Import Rootreducer Đã TạO Ở BưỚC 2.2

```
import { createStore } from "redux";
import { Provider } from "react-redux";

import rootReducer from "./redux/reducers/rootReducer";
```

4. TạO ĐiỂN Store Và TruyỀN VàO Provider

```
const store = creatstore(
  rootReducer,
  window.__REDUX_DEVTOOLS_EXTENSION__ && window.__REDUX_DEVTOOLS_EXTENSION__()
);// window.__REDUX Dùng để mở extension redux devtools trên chrome.

ReactDOM.render(
  <Provider store={store}>
    <App />
  </Provider>,
  document.getElementById("root")
);
```

Redux & Reactjs

TưƠng TÁC Component Với Store

LấY Dự LiệU

Mapstatetoprops

```
import React, { Component } from "react";
import Axios from "./Axios";
import { connect } from "react-redux";

class Users extends Component {
  constructor() {
    this.state = {
      userlist: this.props.userlist.filter(item => {
        return item[this.state.keyword].indexOf(this.state.keyword.toLowerCase()) !== -1;
      })
    }

    this.deleteuserlist.map((item, index) => {
      return this.state.userlist[index] = users[item] /*
    })
  }

  render() {
    return (
      <div>
        <table class="table">
          <thead>
            <tr>
              <th>Name</th>
              <th>Username</th>
              <th>Email</th>
              <th>Phone Number</th>
              <th>Type</th>
            </tr>
          </thead>
          <tbody>{this.renderHTML()}</tbody>
        </table>
      </div>
    );
  }
}

const mapStateToProps = state => {
  return {
    userlist: state.UserReducer.userlist,
    keyword: state.UserReducer.keyword
  }
};

export default connect(mapStateToProps,null)(Users);
```

State.User reducer Trong MapStatetoprops Phải ĐặT TÊN GiỐng Userreducer Trong Rootreducer

GỌI DỮ LiỆU LÊN Store (Dispatch Action)

Mapdispatchtoprops

```
import React, { Component } from "react";
import { connect } from "react-redux";
import * as action from "../redux/action/index";

class Search extends Component {
  render() {
    return (
      <input
        type="text"
        className="form-control mb-3 w-50"
        onChange={event => {
          this.props.setFilter(event.target.value);
        }}
        />
      );
  }
}

const mapDispatchToProps = dispatch => {
  return {
    setFilter: keyword => {
      dispatch({
        type: actionType.FILTER,
        keyword
      })
    }
  }
};

export default connect( null,mapDispatchToProps)(Search);
```

LuÔN PHẢI CÓ HÀM Connect ĐỐC LÊn Component, Tham SỐ 1 CỦA HÀM Connect LÀ Mapstatetoprops (LẤY DỮ LiỆU) Tham SỐ 2 LÀ (Mapdispatchtoprops). KhÔng ĐỂ Mapstatetoprops Ở Tham SỐ 2 VÀ NgƯỜC Lại

xỬ LÝ Ở Reduser Con

```
import * as ActionType from "../../constants/ActionType";

let initState = {
  userlist: [...
  ],
  userEdit: null,
  keyword: ""
};

const userReducer = (state = initState, action) => {
  switch (action.type) {
    case ActionType.FILTER:
      state.keyword = action.keyword;
      return { ...state };
    default:
      return { ...state };
  }
};

export default userReducer;
```

Userreducer Sẽ NhậN Action TruyỀn LÊN NêU ĐúNg Action.Type Thì Nó Sẽ Sao ChéP HiỆT QuÁ Và TrẢ VỀ State

TruyỀn Cho State Bên ĐƯỢC Của Userreducer Là Initstate