## Page:1

# React-Redux CRUD Operation

## Main.jsx

import ReactDOM from "react-dom/client";

import App from "./App.jsx";

import { Provider } from "react-redux";

import { BrowserRouter } from "react-router-dom";

import { store } from "./Redux/store.js";

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

<Provider store={store}>

<BrowserRouter>

<App />

</BrowserRouter>

</Provider>

);

## App.jsx

import MainRoutes from "./MainRoutes";

import Navbar from "./Components/Navbar";

function App() {

return (

<>

<Navbar />

<MainRoutes />

</>

);}

export default App;

## Page:2

## db.json

## package.json

"dependencies": {

"axios": "^1.7.2",

"json-server": "^0.17.4",

"react": "^18.3.1",

"react-dom": "^18.3.1",

"react-redux": "^9.1.2",

"react-router-dom": "^6.24.1",

"redux": "^5.0.1",

"redux-thunk": "^3.1.0"

},

## MainRoutes.jsx

import { Route, Routes } from "react-router-dom";

import Login from "./pages/Login";

import Admin from "./Components/Admin";

import Homepage from "./Components/Homepage";

import Edit from "./pages/Edit";

// import PrivateRoutes from "./Components/PrivateRoutes";

const MainRoutes = () => {

return (

<div>

<Routes>

<Route path="/login" element={<Login />}></Route>

<Route

path="/"

element={

// <PrivateRoutes>

<Homepage />

## Page:3

// </PrivateRoutes>

}

></Route>

<Route path="\*" element={<h1>404 Page Not Found</h1>}></Route>

<Route path="/admin" element={<Admin />}></Route>

<Route path="/edit/:id" element={<Edit />}></Route>

</Routes>

</div>

);

};

export default MainRoutes;

# Redux Folder

# Folders

## Auth

## =>Files

## 1. action.js

## 2. reducer.js

## Product

## =>Files

## 1. action.js

## 2. reducer.js

## =>Files

## actiontype.js

## store.js

## Page:4

## store.js

import { applyMiddleware, combineReducers, legacy\_createStore } from "redux";

import { thunk } from "redux-thunk";

import { reducer as ProductReducer } from "./Products/reducer";

import { reducer as AuthReducer } from "./Auth/reducer";

const rootReducer = combineReducers({

ProductReducer,

AuthReducer,

});

export const store = legacy\_createStore(rootReducer, applyMiddleware(thunk));

## actiontype.js

//Product

export const POST\_DATA\_REQUEST = "POST\_DATA\_REQUEST";

export const POST\_DATA\_SUCCESS = "POST\_DATA\_SUCCESS";

export const POST\_DATA\_FAIL = "POST\_DATA\_FAIL";

//Auth

export const AUTH\_DATA\_REQUEST = "AUTH\_DATA\_REQUEST";

export const AUTH\_DATA\_SUCCESS = "AUTH\_DATA\_SUCCESS";

export const AUTH\_DATA\_FAIL = "AUTH\_DATA\_FAIL";

//GET

export const GET\_DATA\_REQUEST = "GET\_DATA\_REQUEST";

export const GET\_DATA\_SUCCESS = "GET\_DATA\_SUCCESS";

export const GET\_DATA\_FAIL = "GET\_DATA\_FAIL";

## Page:5

# Auth Folder

## action.js

import axios from "axios";

import {

AUTH\_DATA\_FAIL,

AUTH\_DATA\_REQUEST,

AUTH\_DATA\_SUCCESS,

} from "../actiontype";

export const postuserdata = (userdata) => (dispatch) => {

dispatch({ type: AUTH\_DATA\_REQUEST, payload: true });

return axios

.post("https://reqres.in/api/login", userdata)

.then((res) => {

console.log(res.data.token),

dispatch({ type: AUTH\_DATA\_REQUEST, payload: false });

dispatch({ type: AUTH\_DATA\_SUCCESS, payload: res.data.token });

})

.catch((err) => {

console.log(err);

dispatch({ type: AUTH\_DATA\_REQUEST, payload: false });

dispatch({ type: AUTH\_DATA\_FAIL, payload: true });

});

};

## Page:6

## Reducer.js

import { AUTH\_DATA\_SUCCESS } from "../actiontype";

import { AUTH\_DATA\_REQUEST } from "../actiontype";

import { AUTH\_DATA\_FAIL } from "../actiontype";

const initialstate = {

isLoading: false,

isError: false,

token: "",

isAuth: false,

};

// eslint-disable-next-line no-unused-vars

export const reducer = (state = initialstate, { type, payload }) => {

switch (type) {

case AUTH\_DATA\_REQUEST:

return { ...state, isLoading: payload };

case AUTH\_DATA\_SUCCESS:

return { ...state, token: payload, isAuth: true };

case AUTH\_DATA\_FAIL:

return { ...state, isError: payload };

default:

return state;

}

};

## Page:7

# Product Folder

## action.js

import axios from "axios";

import {

GET\_DATA\_FAIL,

GET\_DATA\_REQUEST,

GET\_DATA\_SUCCESS,

POST\_DATA\_FAIL,

POST\_DATA\_REQUEST,

POST\_DATA\_SUCCESS,

} from "../actiontype";

export const postDatafun = (data) => (dispatch) => {

dispatch({ type: POST\_DATA\_REQUEST, payload: true });

axios

.post("http://localhost:8080/product", data)

.then((res) => {

console.log(res);

dispatch({ type: POST\_DATA\_REQUEST, payload: false });

dispatch({ type: POST\_DATA\_SUCCESS, payload: res.data });

})

.catch((err) => {

console.log(err);

dispatch({ type: POST\_DATA\_REQUEST, payload: false });

dispatch({ type: POST\_DATA\_FAIL, payload: true });

});

};

//

export const getdataFromServer = (paramobj) => (dispatch) => {

## Page:8

dispatch({ type: GET\_DATA\_REQUEST, payload: true });

axios

.get("http://localhost:8080/product", paramobj)

.then((res) => {

console.log(res);

// console.log(res.headers["x-total-count"]);

dispatch({ type: GET\_DATA\_REQUEST, payload: false });

dispatch({

type: GET\_DATA\_SUCCESS,

payload: { data: res.data, page: res.headers["x-total-count"] },

});

})

.catch((err) => {

console.log(err);

dispatch({ type: GET\_DATA\_REQUEST, payload: false });

dispatch({ type: GET\_DATA\_FAIL, payload: true });

});

};

## reducer.js

import {

GET\_DATA\_FAIL,

GET\_DATA\_REQUEST,

GET\_DATA\_SUCCESS,

POST\_DATA\_FAIL,

POST\_DATA\_REQUEST,

POST\_DATA\_SUCCESS,

} from "../actiontype";

const initialstate = {

## Page:9

isLoading: false,

isError: false,

data: [],

totalpage: 1,

};

// eslint-disable-next-line no-unused-vars

export const reducer = (state = initialstate, { type, payload }) => {

switch (type) {

case POST\_DATA\_REQUEST:

return { ...state, isLoading: payload };

case POST\_DATA\_SUCCESS:

return { ...state };

case POST\_DATA\_FAIL:

return { ...state, isError: payload };

//GET

case GET\_DATA\_REQUEST: {

return { ...state, isLoading: payload };

}

case GET\_DATA\_SUCCESS: {

return { ...state, data: payload.data, totalpage: payload.page };

}

case GET\_DATA\_FAIL: {

return { ...state, isError: payload };

}

default:

return state;

}

## Page:10

};

# Components

## 1.Admin.jsx

## 2.Homepage.jsx

## 3.Navbar.jsx

## 4.PrivateRoutes.jsx

## 5.Product.jsx

## 6.SIdebar.jsx

## Admin.jsx

import { useState } from "react";

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

import { postDatafun } from "../Redux/Products/action";

const initialstate = {

title: "",

price: "",

description: "",

category: "",

image: "",

};

const Admin = () => {

const [data, setData] = useState(initialstate);

const dispatch = useDispatch();

const postdata = useSelector((store) => store.ProductReducer);

console.log(postdata);

const { title, price, description, category, image } = data;

const handleChange = (e) => {

## Page:11

setData({ ...data, [e.target.name]: e.target.value });

};

const handleSubmit = (e) => {

e.preventDefault();

console.log(data);

dispatch(postDatafun(data));

};

return (

<div>

<h1>Product</h1>

<form action="" onSubmit={handleSubmit}>

<input

type="text"

onChange={(e) => handleChange(e)}

name="title"

value={title}

placeholder="Title..."

/>

<br />

<br />

<input

type="text"

onChange={(e) => handleChange(e)}

name="image"

value={image}

placeholder="Image..."

/>

<br />

<br />

## Page:12

<input

type="text"

onChange={(e) => handleChange(e)}

name="price"

value={price}

placeholder="Price..."

/>

<br />

<br />

<input

type="text"

onChange={(e) => handleChange(e)}

name="description"

value={description}

placeholder="Description..."

/>

<br />

<br />

<select

name="category"

onChange={(e) => handleChange(e)}

value={category}>

<option hidden>Select Your Catrgory</option>

<option value="men's clothing">Men</option>

<option value="women's clothing">Women</option>

<option value="jewelery">Jewelery</option>

<option value="electronics">Electronics</option>

</select>

<br />

<br />

## Page:13

<input type="submit" />

</form>

</div>

);};

export default Admin;

## Homepage.jsx

import Product from "./Product";

import Sidebar from "./Sidebar";

const Homepage = () => {

return (

<div>

<div style={{ display: "flex" }}>

<div style={{ width: "20%", marginTop: "10px" }}>

<Sidebar />

</div>

<div style={{ width: "80%" }}>

<Product />

</div>

</div>

</div>

);

};

export default Homepage;

## Navbar.jsx

import { Link } from "react-router-dom";

const Navbar = () => {

return (

<div style={{ display: "flex", justifyContent: "space-around" }}>

## Page:14

<Link to={"/admin"}>Admin</Link>

<Link to={"/login"}>Login</Link>

</div>

);

};

export default Navbar;

## PrivateRoutes.jsx

import { useSelector } from "react-redux";

import { Navigate } from "react-router-dom";

// eslint-disable-next-line react/prop-types

const PrivateRoutes = ({ children }) => {

const isAuth = useSelector((s) => s.AuthReducer.isAuth);

if (!isAuth) {

return <Navigate to={"/login"} />;

}

return children;

};

export default PrivateRoutes;

## Product.jsx

import { useEffect, useState } from "react";

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

import { Link, useSearchParams } from "react-router-dom";

import { getdataFromServer } from "../Redux/Products/action";

import axios from "axios";

const Product = () => {

## Page:15

const productdata = useSelector((store) => store.ProductReducer);

console.log(productdata);

const { isLoading, isError, data, totalpage } = useSelector(

(store) => store.ProductReducer

);

// console.log(totalpage);

const dispatch = useDispatch();

// eslint-disable-next-line no-unused-vars

const [search, setSearch] = useSearchParams();

// console.log(search.getAll("category"));

const [page, setPage] = useState(1);

const [find, setFind] = useState("");

useEffect(() => {

let paraombj = {

params: {

// category: [

// "jewelery",

// "electronics",

// "men's clothing",

// "women's clothing",

// ],

category: search.getAll("category"),

\_sort: "price",

\_limit: 5,

## Page:16

\_page: page,

\_order: search.get("sort"),

q: find,

},

};

let id = setTimeout(() => {

dispatch(getdataFromServer(paraombj));

}, 1000);

return () => {

console.log("clean up");

clearTimeout(id);

};

// eslint-disable-next-line react-hooks/exhaustive-deps

}, [search, page, find]);

const handleDelete = (id) => {

axios

.delete(`http://localhost:8080/product/${id}`)

.then((res) => {

console.log(res);

dispatch(getdataFromServer({}));

})

.catch((err) => console.log(err));

};

return isLoading ? (

<h1 style={{ color: "#2980B9" }}>Loading...</h1>

) : isError ? (

<h1 style={{ color: "red" }}>Something Wrong</h1>

) : (

## Page:17

<div>

<div style={{ textAlign: "center" }}>

<input

type="text"

onChange={(e) => setFind(e.target.value)}

placeholder="search here..."

/>

</div>

<div style={{ textAlign: "right", margin: "10px" }}>

<button disabled={page == 1} onClick={() => setPage(page - 1)}>

Prev

</button>

<button disabled>{page}</button>

<button

disabled={page == totalpage / 4}

onClick={() => setPage(page + 1)}

>

Next

</button>

</div>

<div

style={{

display: "grid",

gridTemplateColumns: "repeat(3,1fr)",

gap: "10px",

}}

>

{data.length > 0 &&

data.map((el) => (

<div

## Page:18

style={{

boxShadow:

"rgba(0, 0, 0, 0.5) 0px 20px 25px -5px, rgba(0, 0, 0, 0.04) 0px 10px 10px -5px",

textAlign: "center",

}}

key={el.id}

>

<img src={el.image} alt="" height={200} width={200} />

<p>{el.title}</p>

<p>{el.price}</p>

<div

style={{

display: "flex",

gap: "5px",

justifyContent: "center",

}}

>

<button onClick={() => handleDelete(el.id)}>Delete</button>

<Link to={`/edit/${el.id}`}>

<button>Edit</button>

</Link>

<br />

</div>

</div>

))}

</div>

</div>

);

};

## Page:19

export default Product;

## Sidebar.jsx

// import { useEffect } from "react";

import { useEffect } from "react";

import { useState } from "react";

import { useSearchParams } from "react-router-dom";

const Sidebar = () => {

//concept of useSearchParamsa

const [search, setSearch] = useSearchParams();

// console.log(search.getAll("gender"));

// useEffect(() => {

// setSearch({ gender: "male", category: "electronics" }); //serachparam object lay key and value

// }, []);

const [category, setCategory] = useState(search.getAll("category") || []);

console.log(search.getAll("category"));

const [sort, setSort] = useState(search.get("sort") || null);

useEffect(() => {

setSearch({ category, sort });

// eslint-disable-next-line react-hooks/exhaustive-deps

}, [category, sort]);

const handleChange = (e) => {

// console.log(e.target.value);

let { value } = e.target;

// console.log(value);

let newcategory = [...category];

## Page:20

if (newcategory.includes(value)) {

newcategory = newcategory.filter((el) => el !== value);

} else {

newcategory.push(value);

}

setCategory(newcategory);

};

return (

<div>

<h1>Category</h1>

<div>

<input

type="checkbox"

value={"men's clothing"}

checked={category.includes("men's clothing")}

onChange={(e) => handleChange(e)}

name=""

id=""

/>

Men

<br />

<br />

<input

type="checkbox"

value={"women's clothing"}

checked={category.includes("women's clothing")}

onChange={(e) => handleChange(e)}

name=""

## Page:21

id=""

/>

Women

<br />

<br />

<input

type="checkbox"

value={"jewelery"}

checked={category.includes("jewelery")}

onChange={(e) => handleChange(e)}

name=""

id=""

/>

Jewelery

<br />

<br />

<input

type="checkbox"

value={"electronics"}

checked={category.includes("electronics")}

onChange={(e) => handleChange(e)}

name=""

id=""

/>

Electronics

</div>

<h2>Sort</h2>

<input type="radio" onChange={() => setSort("asc")} name="price" />

Low To High <br /> <br />

## Page:22

<input type="radio" onChange={() => setSort("desc")} name="price" />

High To Low

</div>

);

};

export default Sidebar;

# Pages Folder

## Files

## 1. Edit.jsx

## 2. Login.jsx

## Edit.jsx

import axios from "axios";

import { useEffect, useState } from "react";

import { useNavigate, useParams } from "react-router-dom";

//populate method

const Edit = () => {

const [price, setPrice] = useState(0);

const { id } = useParams();

// console.log(id);

const navigate = useNavigate();

useEffect(() => {

axios

.get(`http://localhost:8080/product/${id}`)

## Page:23

.then((res) => setPrice(res.data.price))

.catch((err) => console.log(err));

// eslint-disable-next-line react-hooks/exhaustive-deps

}, []);

const handleupdate = (id) => {

axios

.patch(`http://localhost:8080/product/${id}`, { price })

.then((res) => {

console.log(res.data);

alert("Price Updated");

navigate("/");

})

.catch((err) => console.log(err));

};

return (

<div>

<h1>Edit</h1>

<input

type="text"

value={price}

onChange={(e) => setPrice(e.target.value)}

placeholder="Update Price"

/>

<button

onClick={() => {

handleupdate(id);

}}

>

## Page:24

UpdatePrice

</button>

</div>

);

};

export default Edit;

## Login.jsx

import { useState } from "react";

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

import { postuserdata } from "../Redux/Auth/action";

import { useNavigate } from "react-router-dom";

const Login = () => {

const [email, setEmail] = useState("");

const [password, setPassword] = useState("");

const dispatch = useDispatch();

const userdata = useSelector((s) => s.AuthReducer); //data collect from store ,s is only name of parameter

console.log(userdata);

const { isLoading, isError, isAuth } = useSelector((s) => s.AuthReducer); //data collect from store ,s is only name of parameter

const navigate = useNavigate();

const handleSubmit = (e) => {

e.preventDefault();

let userdata = {

email,

## Page:25

password,

};

console.log(userdata);

// eslint-disable-next-line no-unused-vars

dispatch(postuserdata(userdata)).then((res) => {

navigate("/");

});

};

return isLoading ? (

<h1 style={{ color: "#85929E" }}>Please Wait...</h1>

) : isError ? (

<h1 style={{ color: "#CB4335" }}>Something Went to Wrong...</h1>

) : (

<div>

{isAuth ? (

<h1 style={{ color: "green" }}>You are Login</h1>

) : (

<h1 style={{ color: "red" }}>Please Login</h1>

)}

<form action="" onSubmit={handleSubmit}>

<input

type="text"

onChange={(e) => setEmail(e.target.value)}

placeholder="Email..."

/>

<br />

<br />

<input

type="text"

## Page:26

onChange={(e) => setPassword(e.target.value)}

placeholder="Password..."

/>

<br />

<br />

<input type="submit" />

</form>

</div>

);

};

export default Login;