1. Create react app

npx create-react-app my-app

1. Install Redux

npm i redux react-redux redux-thunk

1. Import Provider and store inside App.js

import { Provider } from "react-redux";

import store from "./store";

1. Wrap the provider and store inside render method.

e.g.

<Provider store={store}>

<React.Fragment>

<Header />

<GridView />

<Footer departments={this.departments} />

</React.Fragment>

</Provider>

1. Create a store(store.jsx)

e.g.

import { createStore, applyMiddleware, compose } from "redux";

import thunk from "redux-thunk";

import rootReducer from "./Reducers";

const initialState = {};

const middleware = [thunk];

const store = createStore(

rootReducer,

initialState,

compose(

applyMiddleware(...middleware),

window.\_\_REDUX\_DEVTOOLS\_EXTENSION\_\_ && window.\_\_REDUX\_DEVTOOLS\_EXTENSION\_\_()

)

);

export default store;

1. Create reducers(new foler)

e.g.

index.jsx

import { combineReducers } from "redux";

import postReducer from "./postReducer";

export default combineReducers({

posts: postReducer

});

postReducers.jsx

import { FETCH\_POSTS, NEW\_POST, FETCH\_POST, PUT\_CUST } from "../Actions/types";

const initialState = {

records: [],

record: {},

customer: {},

response: {},

loading: true

};

//Evaluate what type to use

export default function(state = initialState, action) {

switch (action.type) {

case FETCH\_POSTS:

console.log(action.payload);

return { ...state, records: action.payload, loading: true };

case FETCH\_POST:

return { ...state, customer: action.payload };

case NEW\_POST:

return { ...state, record: action.payload };

case PUT\_CUST:

return { ...state, records: action.payload };

default:

return state;

}

}

1. Create Actions

e.g

types.jsx

export const FETCH\_POSTS = "FETCH\_POSTS";

export const FETCH\_POST = "FETCH\_POST";

export const NEW\_POST = "NEW\_POST";

export const PUT\_CUST = "PUT\_CUST";

postActions.jsx

import { FETCH\_POSTS, NEW\_POST, FETCH\_POST, PUT\_CUST } from "./types";

export const fetchCustomers = () => dispatch => {

console.log("fetching..");

fetch("https://localhost:5001/api/customers")

.then(res => res.json())

.then(customers =>

dispatch({

type: FETCH\_POSTS,

payload: customers

})

);

};

export const fetchCustomer = id => dispatch => {

console.log("fetching..");

fetch("https://localhost:5001/api/customers/" + id)

.then(res => res.json())

.then(customer =>

dispatch({

type: FETCH\_POST,

payload: customer

})

);

};

export const createCustomer = customer => dispatch => {

fetch("https://localhost:5001/api/customers", {

method: "POST",

headers: { "Content-Type": "application/json" },

body: JSON.stringify(customer)

})

.then(res => res.json())

.then(post =>

dispatch({

type: NEW\_POST,

payload: post

})

);

};

export const modifyCustomer = customer => dispatch => {

fetch("https://localhost:5001/api/customers/" + customer["customerKey"], {

method: "PUT",

headers: { "Content-Type": "application/json" },

body: JSON.stringify(customer)

})

.then(res => res.json())

.then(post =>

dispatch({

type: PUT\_CUST,

payload: post

})

);

};