REACT REDUX HOMEWORK

Creo mis action-types en /action/action-types.js

export const ADD\_MOVIE\_FAVORITE = "add\_movie\_favorite";

export const GET\_MOVIES = "get\_movies";

export const REMOVE\_MOVIE\_FAVORITE = "remove\_movie\_favorite";

export const GET\_MOVIE\_DETAIL = "get-movie-detail";

CREO MIS ACTIONS EN /ACTION/INDEX.JS

import {

  ADD\_MOVIE\_FAVORITE,

  REMOVE\_MOVIE\_FAVORITE,

  GET\_MOVIES,

  GET\_MOVIE\_DETAIL,

} from "./action-types";

const API\_KEY = "e339fd3e"

export const addMovieFavorite = (payload) => ({

  type: ADD\_MOVIE\_FAVORITE,

  payload,

});

export const removeMovieFavorite = (payload) => ({

  type: REMOVE\_MOVIE\_FAVORITE,

  payload,

});

export const getMovies = (titulo) => {

  return async function (dispatch) {

    const response = await fetch(`http://www.omdbapi.com/?apikey=${API\_KEY}&s=${titulo}`);

    const json = await response.json();

    dispatch({ type: GET\_MOVIES, payload: json });

  };

};

export const getMovieDetail = (payload) => ({

  type: GET\_MOVIE\_DETAIL,

  payload,

});

CREO MI INITIAL STATE EN /REDUCERS/INDEX.JS

const initialState = {

  moviesFavourites: [],

  moviesLoaded: [],

  movieDetail: {},

};

CREO MI FUNCTION REDUCER EN /REDUCERS/INDEX.JS

import {

  ADD\_MOVIE\_FAVORITE,

  REMOVE\_MOVIE\_FAVORITE,

  GET\_MOVIES,

  GET\_MOVIE\_DETAIL,

} from "../actions/action-types";

const initialState = {

  moviesFavourites: [],

  moviesLoaded: [],

  movieDetail: {},

};

const rootReducer = (state = initialState, { type, payload }) => {

  switch (type) {

    case ADD\_MOVIE\_FAVORITE:

      return {

        ...state,

        moviesFavourites: state.moviesFavourites.concat(payload),

      };

    case REMOVE\_MOVIE\_FAVORITE:

      return {

        ...state,

      };

    case GET\_MOVIES:

      return {

        ...state,

        moviesLoaded: payload.Search,

      };

    case GET\_MOVIE\_DETAIL:

      return {

        ...state,

      };

    default:

      return state;

  }

};

export default rootReducer;

CREO MI STORE

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

import rootReducer from "../reducers/index";

import thunk from "redux-thunk";

const composeEnhancers = window.\_\_REDUX\_DEVTOOLS\_EXTENSION\_COMPOSE\_\_ || compose;

export const store = createStore(

  rootReducer,

  composeEnhancers(applyMiddleware(thunk))

);

CONECTO MI STORE CON LA APLICACION EN INDEX

import React from "react";

import ReactDOM from "react-dom";

import "./index.css";

import App from "./App";

import \* as serviceWorker from "./serviceWorker";

import { Provider } from "react-redux";

import { store } from "./store";

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

ReactDOM.render(

  <Provider store={store}>

    <BrowserRouter>

      <App />

    </BrowserRouter>

  </Provider>,

  document.getElementById("root")

);

CONECTO MIS COMPONENTES POR AHORA SOLO GET MOVIE

import React, { Component } from "react";

import { connect } from "react-redux";

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

import { addMovieFavorite } from "../../actions";

import { getMovies } from "../../actions";

import "./Buscador.css";

export class Buscador extends Component {

  constructor(props) {

    super(props);

    this.state = {

      title: "",

    };

  }

  handleChange(event) {

    this.setState({ title: event.target.value });

  }

  handleSubmit(event) {

    event.preventDefault();

    //aca lo cambio

    this.props.getMovies(this.state.title)

  }

  render() {

    const { title } = this.state;

    return (

      <div>

        <h2>Buscador</h2>

        <form className="form-container" onSubmit={(e) => this.handleSubmit(e)}>

          <div>

            <label className="label" htmlFor="title">

              Película:{" "}

            </label>

            <input

              type="text"

              id="title"

              autoComplete="off"

              value={title}

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

            />

          </div>

          <button type="submit">BUSCAR</button>

        </form>

        <ul>

          {/\* Aqui tienes que escribir tu codigo para mostrar la lista de peliculas \*/}

          this.

          <li>uno</li>

          <li>dos</li>

          <li>tres</li>

        </ul>

      </div>

    );

  }

}

const mapStateToProps = (state) => ({ movies: state.moviesLoaded });

const mapDispatchToProps = (dispatch) => ({

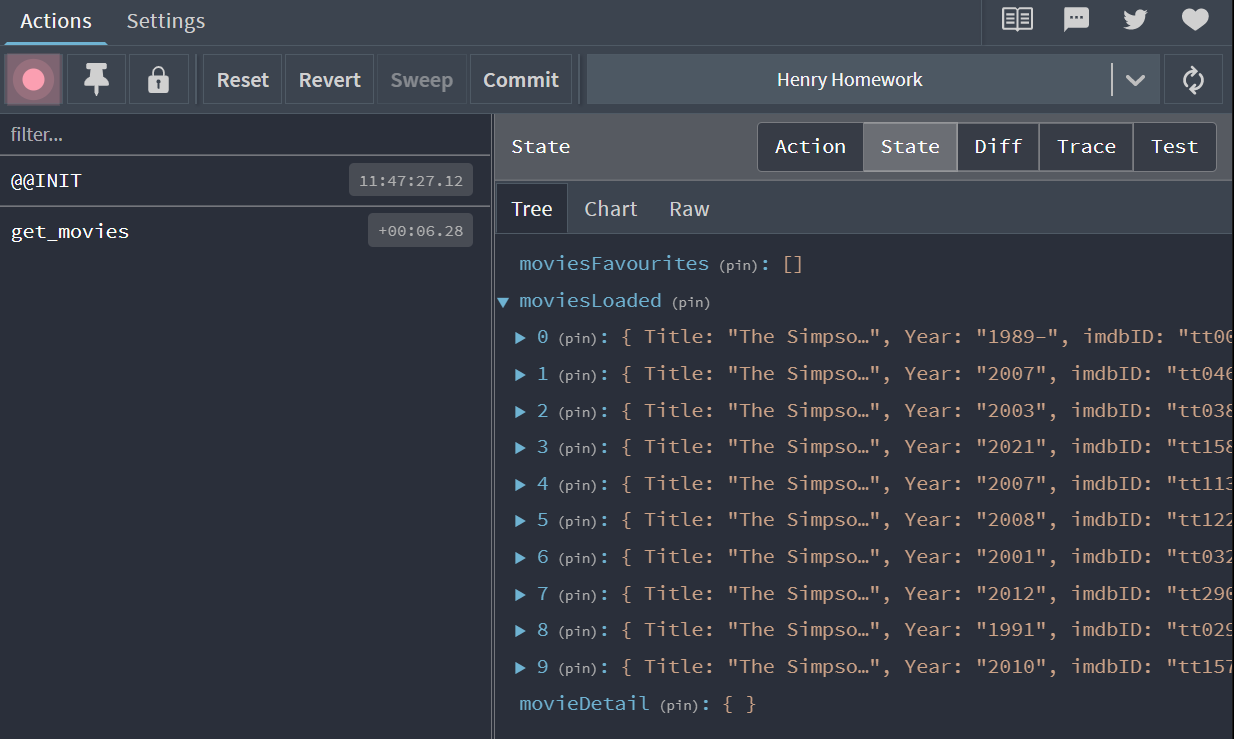
  addMovieFavorite: (movie) => dispatch(addMovieFavorite(movie)),

  getMovies: (title) => dispatch(getMovies(title)),

});

export default connect(mapStateToProps, mapDispatchToProps)(Buscador);

Y EN DEVTOOLS CUANDO PRUEBO YA ME SALE



**YA PUEDO CREAR MI LISTA DE PELIS**

        <ul>

          {/\* Aqui tienes que escribir tu codigo para mostrar la lista de peliculas \*/}

          {this.props.movies.map((movie) => (

            <li key={movie.imdbID}>

              {movie.Title}

              <button>FAV</button>

            </li>

          ))}

        </ul>

**SIGUIENTE PASO, NECESITO PODER AGREGAR MI FAVORITO A FAVORITOS.**

**MI REDUCER:**

    case ADD\_MOVIE\_FAVORITE:

      return {

        ...state,

        moviesFavourites: [...state.moviesFavourites, payload],

      };

**MI ACTION**

export const addMovieFavorite = (movie) => ({

  type: ADD\_MOVIE\_FAVORITE,

  payload: movie,

});

**CONECTO CON MI BOTON EN BUSCADOR.JS**

<ul>

          {/\* Aqui tienes que escribir tu codigo para mostrar la lista de peliculas \*/}

          {this.props.movies.map((movie) => (

            <li key={movie.imdbID}>

              {movie.Title}

              <button onClick={() => this.props.addMovieFavorite(movie)}>

                FAV

              </button>

            </li>

          ))}

        </ul>

**AHORA QUIERO DESHABILITAR EL BOTON CUANDO YA AGREGUE LA PELI A FAVS**

**Agrego favs a mi mapStateToProps en Buscador.js favs, es el listado de favoritos. array**

const mapStateToProps = (state) => ({

  movies: state.moviesLoaded,

  favs: state.moviesFavourites,

});

**Agrego al botón:**

              <button

                onClick={() => this.props.addMovieFavorite(movie)}

                disabled={this.props.favs.find(({imdbID})=>imdbID === movie.imdbID)}>

                FAV

              </button>

**ME QUEDAN DOS PAGINAS, DETALLE DE LA PELICULA Y LA LISTA DE FAVORITOS EN LA PESTANA FAVORITAS**

**MODIFICO COMPONENTS/FAVORITES.JS**

import React, { Component } from "react";

import { connect } from "react-redux";

// import { Link } from 'react-router-dom';

import "./Favorites.css";

import { removeMovieFavorite } from "../../actions";

export class ConnectedList extends Component {

  render() {

    return (

      <div>

        <h2>Películas Favoritas</h2>

        <ul>

          {/\* Aqui tienes que escribir tu codigo para mostrar la lista de peliculas \*/}

          {this.props.favs.map((movie) => (

            <li key={movie.imdbID}>

              {movie.Title}

              <button onClick={() => this.props.removeMovieFavorite(movie.imdbID)}>

                X

              </button>

            </li>

          ))}

        </ul>

      </div>

    );

  }

}

const mapStateToProps = (state) => ({

  favs: state.moviesFavourites,

});

const mapDispatchToProps = {

  removeMovieFavorite,

};

export default connect(mapStateToProps, mapDispatchToProps)(ConnectedList);

**MODIFICO MI ACTION**

export const removeMovieFavorite = (id) => ({

  type: REMOVE\_MOVIE\_FAVORITE,

  payload: id,

});

**MODIFICO MI REDUCER**

    case REMOVE\_MOVIE\_FAVORITE:

      return {

        ...state,

        moviesFavourites: state.moviesFavourites.filter(({ imdbID }) => imdbID!==payload),

      };

**PARA EL DETALLE PODRIA HACER UN ESTADO LOCAL SIMPLEMENTE CON useState**

**CREO LA ACCION DE LA MISMA FORMA QUE CREE LA LISTA EN ACTIONS/INDEX.JS**

export const getMovieDetail = (id) => {

  return async function (dispatch) {

    const response = await fetch(

      `http://www.omdbapi.com/?apikey=${API\_KEY}&i=${id}`

    );

    const json = await response.json();

    dispatch({ type: GET\_MOVIE\_DETAIL, payload: json });

  };

};

**ARREGLO EL COMPONENTE MOVIE.JS**

import React from "react";

import { connect } from "react-redux";

import { getMovieDetail } from "../../actions/index";

import "./Movie.css";

class Movie extends React.Component {

  componentDidMount() {

    const { id } = this.props.match.params;

    this.props.getMovieDetail(id);

  }

  render() {

    return <div className="movie-detail">Detalle de la pelicula</div>;

  }

}

const mapStateToProps = (state) => ({

});

const mapDispatchToProps = {

  getMovieDetail,

};

export default connect(mapStateToProps, mapDispatchToProps)(Movie);

**LE AGREGO LINK DE REACT-DOM PARA QUE DE EL DETALLE EN BUSCADOR**

<ul>

          {/\* Aqui tienes que escribir tu codigo para mostrar la lista de peliculas \*/}

          {this.props.movies.map((movie) => (

            <li key={movie.imdbID}>

              <Link to={`/movie/${movie.imdbID}`}>{movie.Title}</Link>

              <button

                onClick={() => this.props.addMovieFavorite(movie)}

                disabled={this.props.favs.find(

                  ({ imdbID }) => imdbID === movie.imdbID

                )}

              >

                FAV

              </button>

            </li>

          ))}

        </ul>

**MIRO EN EL INSPECTOR DEL NAVEGADOR LO Q LLEGA Y ES UN OBJETO, ESE ES MI PAYLOAD. CON ESO HAGO MI REDUCER.**

    case GET\_MOVIE\_DETAIL:

      return {

        ...state,

        movieDetail: payload,

      };

**ORGANIZO MI RENDER EN MOVIE.JS**

import React from "react";

import { connect } from "react-redux";

import { getMovieDetail } from "../../actions/index";

import "./Movie.css";

class Movie extends React.Component {

  componentDidMount() {

    const { id } = this.props.match.params;

    this.props.getMovieDetail(id);

  }

  render() {

    return (

      <div className="movie-detail">

        Detalle de la pelicula

        <h2>{`Titulo: ${this.props.movies.Title}`}</h2>

        <img src={this.props.movies.Poster} alt='Img not found'/>

        <h4>{`Año: ${this.props.movies.Year}`}</h4>

        <h4>{`Duración: ${this.props.movies.Runtime}`}</h4>

        <h4>{`Elenco: ${this.props.movies.Actors}`}</h4>

        <h4>{`Director: ${this.props.movies.Director}`}</h4>

      </div>

    )

  }

}

const mapStateToProps = (state) => ({

  movies: state.movieDetail,

});

const mapDispatchToProps = {

  getMovieDetail,

};

export default connect(mapStateToProps, mapDispatchToProps)(Movie);

**ACA YA FUNCIONA TODO BIEN PERO ESTA MUY LENTA Y SE DEMORA CARGANDO**

const initialState = {

  moviesFavourites: [],

  moviesLoaded: [],

  movieDetail: undefined,

};

**MODIFICO EL RENDER DE MOVIE.JS**

render() {

    return (

      <div className="movie-detail">

        Detalle de la pelicula

        {this.props.movies === undefined ? (

          <>

            <h2>Cargando...</h2>

          </>

        ) : (

          <>

            <h2>{`Titulo: ${this.props.movies.Title}`}</h2>

            <img src={this.props.movies.Poster} alt="Img not found" />

            <h4>{`Año: ${this.props.movies.Year}`}</h4>

            <h4>{`Duración: ${this.props.movies.Runtime}`}</h4>

            <h4>{`Elenco: ${this.props.movies.Actors}`}</h4>

            <h4>{`Director: ${this.props.movies.Director}`}</h4>

          </>

        )}

      </div>

    );

  }

}

**CAMBIO LA ACTION DE GETMOVIEDETAIL**

export const getMovieDetail = (id) => {

  return async function (dispatch) {

    if (id) {

      try {

        const response = await fetch(

          `http://www.omdbapi.com/?apikey=${API\_KEY}&i=${id}`

        );

        const json = await response.json();

        if (json.Response === "False") {

          dispatch({ type: GET\_MOVIE\_DETAIL, payload: null });

        } else {

          dispatch({ type: GET\_MOVIE\_DETAIL, payload: json });

        }

      } catch {

        dispatch({ type: GET\_MOVIE\_DETAIL, payload: null });

      }

    } else {

      dispatch({ type: GET\_MOVIE\_DETAIL });

    }

  };

};

**AGREGO EN MOVIE.JS**

  componentWillUnmount() {

    this.props.getMovieDetail();

  }