Tipo : Guía de laboratorio
Capítulo : React JS
Duración : 90 minutos

I. OBJETIVO

Crear un proyecto React integrado con Redux para mejorar el manejo de la información en la aplicación web.

II. REQUISITOS

Los siguientes elementos de software son necesarios para la realización del laboratorio:

- Editor Visual Studio Code
- Node JS

III. EJECUCIÓN DEL LABORATORIO

- **Ejercicio 2.7: Utilizar Redux para manejar la información entre componentes**

1. Instalar la librería Redux y sus dependencias.

```
npm install --save redux react-redux redux-logger redux-thunk
```

2. Crear el store de la aplicación.

createStore.js

```
src > store > createStore.js > ...
1  import { applyMiddleware, createStore } from 'redux';
2  import { logger } from 'redux-logger';
3  import thunk from 'redux-thunk';
4
5  import reducers from '../reducers';
6
7  const middlewares = process.env.NODE_ENV === 'development' ? [logger] : [];
8
9  export default createStore(
10     reducers,
11     applyMiddleware(thunk, ...middlewares)
12 );
```

3. Crear el reducer de productos.

productReducer.js

src > reducers > productReducer.js > ...

```
} from '../constasts/action-types';

const initialState = {
    items: [],
    item: {},
    filtered: []
};

export default (state = initialState, { type, payload }) => {
    switch(type) {
        case FETCH_PRODUCTS_SUCCESS:
            return { ...state, items: payload, item: {}, filtered: [] };
        case FETCH_PRODUCT_SUCCESS:
            return { ...state, item: payload };
        case ADD_PRODUCT_SUCCESS:
            return { ...state, items: [...state.items, payload ] };
        case REMOVE_PRODUCT_SUCCESS:
            return { ...state, items: state.items.filter(product =>product.id !== payload) }
        case FILTER_PRODUCTS:
            const value = payload.toLowerCase();
            return { ...state, filtered: state.items.filter(({ name }) => {
                return name.toLowerCase().includes(value);
            })};
        default:
            return state;
    }
};
```

4. Crear las acciones de productos

productAction.js

src > actions > productAction.js > ...

```
export const fetchProducts = () => {
    return dispatch => {
        ProductService.getProducts()
            .then(response => {
                dispatch({ type: FETCH_PRODUCTS_SUCCESS, payload: response.data });
            });
    };
};

export const fetchProduct = (id) => {
    return async dispatch => {
        const response = await ProductService.getProduct(id);

        dispatch({ type: FETCH_PRODUCT_SUCCESS, payload: response.data });
    }
}

export const createProduct = product => {
    return dispatch => {
        ProductService.createProduct(product)
            .then(response => {
                dispatch({ type: ADD_PRODUCT_SUCCESS, payload: response.data });
            });
    };
};

```

5. Conectar el componente Products a Redux.

src/modules/Products.js

```
const mapStateToProps = state => {
    return {
        products: state.products.items,
        filtered: state.products.filtered
    }
};

const mapDispatchToProps = dispatch => {
    return {
        getProducts: () => {
            dispatch(fetchProducts());
        },
        removeProduct: (id) => {
            dispatch(removeProduct(id));
        },
        filterProducts: (searchValue) => {
            dispatch(filterProducts(searchValue));
        }
    };
};

export default withRouter(
    connect(mapStateToProps, mapDispatchToProps)(Products)
);
```

.

6. Invocar a las acciones desde el componente.

src/modules/Products.js

```
    componentDidMount() {
        this.props.getProducts();
    }

    handleDelete = id => {
        this.props.removeProduct(id);
    };

    handleSearchProduct = searchValue => {
        this.props.filterProducts(searchValue);
    };
```

IV. EVALUACIÓN

1. ¿Cuándo usar Redux?

   Redux se usa para proyectos grandes en donde se necesita comunicar mucho el estado entre componentes. Con Redux se logra que la aplicación sea más escalable y fácil de mantener.

2. Si no se usara Redux, ¿cómo comunicarías la data entre componentes?

   Pasando la data por los props de un componente padre a un componente hijo.