

React.JS

Introducción

Módulo 04

Redux Planificación

Planificar un store

Redux es una librería que ofrece herramientas muy escalables. Sin embargo, es muy fácil perderse. Son muchos archivos distintos, comunicados entre sí. Sin un buen modelo del Store, es muy fácil perdernos entre las líneas de código, incluso en proyectos pequeños.

Para planificar un Store, vamos a modelar lo siguiente:

- El estado global de toda la aplicación.
- Los cambios de estado que deben ocurrir en la aplicación.
- Las lecturas de ese estado global.



Modelando el estado global

Modelar el estado es la parte menos abstracta, pero, quizás, la más tediosa.

```
mockStore.json ×
src > {} mockStore.json > ...
1  {
2    "users": {
3      "currentUser": {
4        "id": 0,
5        "name": "cvallejos"
6      }
7    },
8    "products": {
9      "productSource": [
10       { "id": 1, "name": "Teclado", "stock": 10 },
11       { "id": 2, "name": "Mouse", "stock": 15 },
12       { "id": 3, "name": "Pendrive", "stock": 5 }
13     ]
14   }
15 }
```

Nos puede ayudar mucho, **crear un archivo JSON con todos los datos que debe manejar la aplicación y un valor por defecto para cada dato.**

La idea central de esta etapa es que logres tener un archivo en donde puedas ver **todo** el estado de tu aplicación. Ese estado conviene que esté dividido en **categorías** de información (como users o products) para luego pasarlo a Redux más fácilmente).

Definiendo los cambios de estado

(Acciones)

Para modelar los cambios de estado te puede ayudar la siguiente tabla:

Nombre	Porción del estado que cambia	Información del cambio	Cómo queda el nuevo estado
USER/SET_USER	user: {...}	nuevoUsuario	user: { currentUser: nuevoUsuario }
PRODUCT/SET_SOURCE	products: {...}	nuevosProductos	products: { productSource: nuevosProductos }

Puede haber muchas formas de modelar las Acciones al estado, pero planificando estos datos de antemano te será mucho más fácil diseñar las acciones en código luego.

Definiendo las lecturas del estado

También puede ser útil pensar las lecturas (o “vistas”) del estado. Ésto te será útil cuando quieras consultar tu estado desde React.

Nombre	Descripción	Parámetros	Porción del estado global que devuelve
GET_CURRENT_USER	Obtener el usuario actual	Ninguno	{ user: { currentUser } }
GET_ONE_PRODUCT	Obtener un solo producto	productId	{ products: { productSource.filter(item => item.id == productId) } }

Planificar un store

Si bien no es estrictamente necesario realizar este proceso, es muy conveniente a la hora de adentrarse a gestionar un Store. ¡Puede ahorrar muchos dolores de cabeza!

Más allá de los modelos, la idea central es muy simple:

Empezar a escribir nuestro Store con una idea clara de lo que queremos construir.

¡Muchas gracias!

¡Sigamos trabajando!