

Evaluación grupal



¿Qué es Redux?

¿Qué es Redux?

Es una librería que nos habilita un data store, el cual permite mover los datos de una aplicación fuera de la jerarquía de componentes, lo que significa que no es necesario elevar los datos y luego ponerlos a disposición de los componentes descendientes a través props.



¿Por qué es útil?



¿Por qué es útil?

Los data stores pueden simplificar los componentes en un proyecto, produciendo una aplicación que sea más fácil de desarrollar y probar.



¿Cómo se usa?

¿Cómo se usa?

Los datos se mueven a una parte dedicada de la aplicación a la que pueden acceder directamente los componentes que lo requieren. En el caso de Redux, los componentes están conectados al data store a través de props, lo que aprovecha la naturaleza de React, aunque el proceso de mapeo en sí mismo puede ser incómodo y requerir mucha atención.



¿Hay alguna limitación?

¿Hay alguna limitación?

Redux agrega rigidez y complejidad a una aplicación. Lo hace de una manera de bajo nivel (modificamos directamente el objeto de estado en sus reducers) y depende de la mentalidad de map/reduce para funcionar. Para ese fin, los inconvenientes de Redux son: Rigidez.

Estos métodos específicos para tratar datos pueden ser considerados restrictivos por algunos desarrolladores.



¿Hay alguna alternativa?



¿Hay alguna alternativa?

No todas las aplicaciones necesitan un data store. Para cantidades más pequeñas de datos, el uso de características de estado de componentes puede ser aceptable y la context API de React, puede usarse para características básicas de administración de datos.



¿Cómo creamos un data store?



¿Cómo creamos un data store?

Definimos los datos iniciales, los action types, los action creators y los reducers.




¿Cómo agregamos un data
store a una aplicación de React?




¿Cómo agregamos un data store a una aplicación de React?

Usamos el componente Provider del paquete React-Redux



¿Cómo consumimos un data store
en un componente de React?



¿Cómo consumimos un data store en un componente de React?

Usamos la función `connect` para mapear las props del componente a los datos del store y los action creators.



¿Qué es una action?



¿Qué es una action?

Una acción describe una operación que cambiará los datos en el store. Redux no permite que los datos se modifiquen directamente y requiere actions para especificar los cambios.



¿Qué es el action type?

¿Qué es el action type?

Las acciones son objetos JavaScript simples que tienen un parámetro type, que especifica el action type. Esto garantiza que las actions se puedan identificar y procesar correctamente.



¿Qué es un action creator?

¿Qué es un action creator?

Un action creator es una función que crea una acción. Los action creators se presentan a los componentes de React como props de funciones para que invocar la función del action creator aplique un cambio al data store.



¿Qué es un reducer?

¿Qué es un reducer?

Un reducer es una función que recibe una action y procesa el cambio que representa en el data store. Una acción especifica qué operación se debe aplicar al data store, pero es el reducer el que contiene el código JavaScript que lo hace posible.