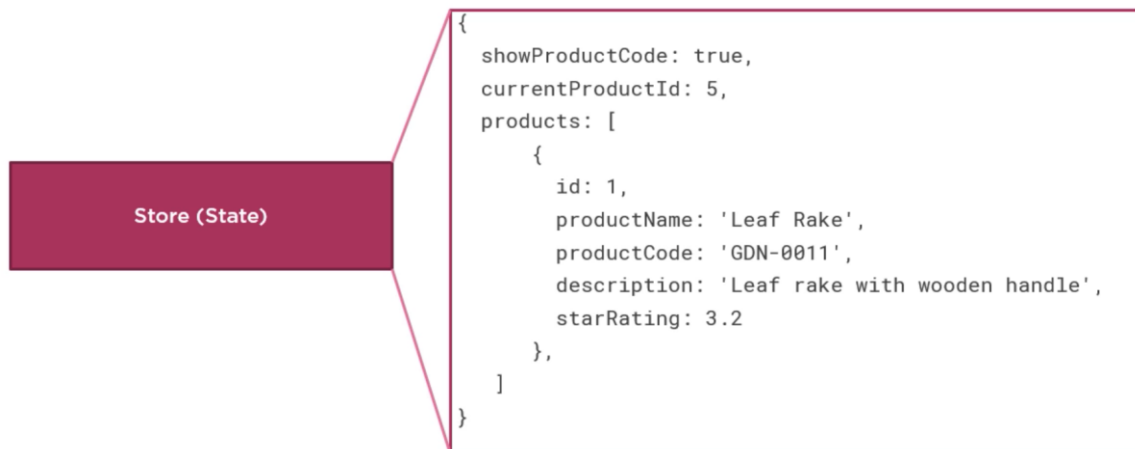


El patrón Redux – Store

El Store es literalmente un objeto JavaScript que contiene todo el estado de su aplicación:



Puede pensar en el Store como una base de datos del lado del cliente.

Con Angular, puede construir servicios para mantener el estado de su aplicación. Siguiendo el patrón Redux, todo ese estado se conserva en el Store. Esto se vuelve muy poderoso cuando se trata de razonar sobre la interacción del usuario, la depuración, el rendimiento y evitar condiciones de error en general.

¿Hay que poner todo en el Store? No, no es necesario

¿Qué es lo que NO se debe poner en el Store?

- Un estado no compartido que es propiedad exclusiva de un único componente que no necesita ser compartido o puesto a disposición a través de las diferentes rutas.
- Los Formularios de Angular tampoco pertenecen a un Store, ya que suelen ser autónomos y no necesitan compartirse entre varios componentes. Además, los Formularios de Angular no son serializables o inmutables y pueden cambiar por sí mismos, lo que significa que no será capaz de rastrear esos cambios usando Actions, que son el segundo principio de Redux: para cambiar el estado, necesita emitir Actions.
- El estado que tiene ciclos en él o tiene estructuras de datos complejas que no pueden ser serializadas no deben ser puestos en el Store. Por ejemplo, todo el estado del Router no debe ser puesto en el Store porque no es serializable.