Redux Store

We now introduce the concept of a Redux Store, which is a representation of the most essential rule of Redux programming: the whole application is handled by one state object.

The bank vault keeps the money in the bank, right?

Well, within your application, you don't spend money. Instead, the STATE of your application is like the money you spend. The entire user interface of your application is a function of your state.

Just like the bank vault keeps your money safe in the bank, the state of your application is kept safe by something called a STORE. So, the STORE keeps your "money" i.e state, intact.

Uh, you need to remember this, okay?

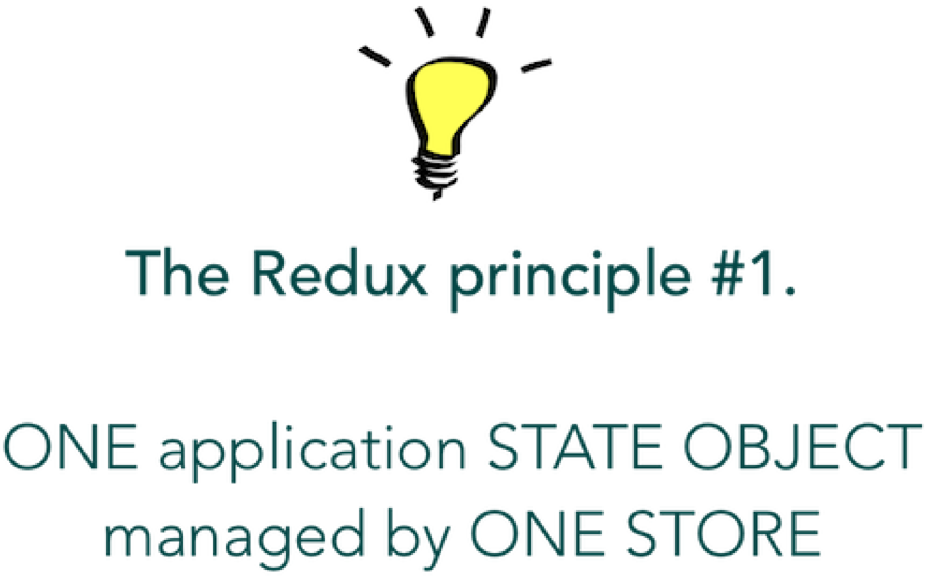
The Redux Store can be likened to the Bank Vault. It holds the state of your application - and keeps it safe.

This leads to the first Redux principle:

> Have a single source of truth: The state of your whole application is stored in an object tree within a single Redux store.

Don’t let the words confuse you.

In simple terms, with Redux, it is is advisable to store your application state in a single object managed by the Redux STORE. It’s like having ONE VAULT as opposed to littering money everywhere along the bank hall.



The Redux principle #1.

ONE application STATE OBJECT
managed by ONE STORE