The Store and Reducer Relationship

The role of the reducer is to help us access the store.

Back to the bank analogy.

When you go to the bank to make a withdrawal, you meet with the Cashier. After you make your WITHDRAW_MONEY intent/action known to the Cashier, they do NOT just hand you the requested money.

No.

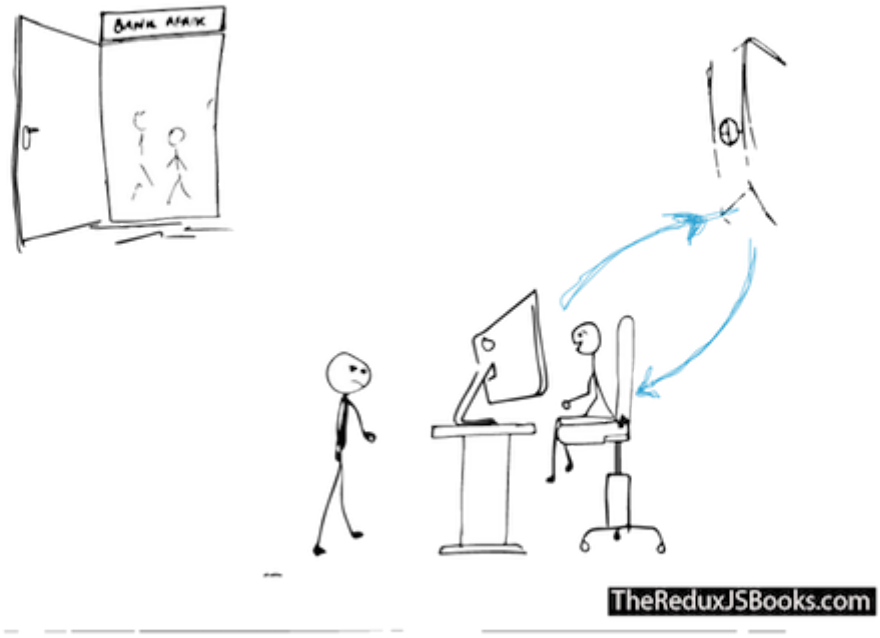
The Cashier first confirms that you have enough money in your account to perform the withdrawal transaction you seek.

The Cashier first makes sure you have the money you say you do.

From the computer, they can see all that - kind of communicating with the

From the computer, they can see all that - kind of communicating with the Vault , since the Vault keeps all the money in the bank.

In a nutshell, the Cashier and Vault are always in sync. Great buddies!



The same may be said for a Redux **STORE** (our own Vault), and the Redux **REDUCER** (our own Cashier)

The Store and the Reducer are great buddies. Always in sync.

Why?

The REDUCER always “talks” to the STORE. Just like the Cashier stays in sync with the Vault.

This explains why the creation of the store needs to be invoked with a Reducer, and that is mandatory. The Reducer is the only mandatory argument passed into createStore():

JS App.js

```
1   import React, { Component } from "react";
2   import HelloWorld from "./HelloWorld";
3
4   import { createStore } from "redux";
5   const store = createStore(reducer);
6
7   class App extends Component {
8     render() {
```

In the following section we will have a brief look at Reducers and then create a STORE by passing the REDUCER into the createStore factory function.