Architecture and Main Characteristics

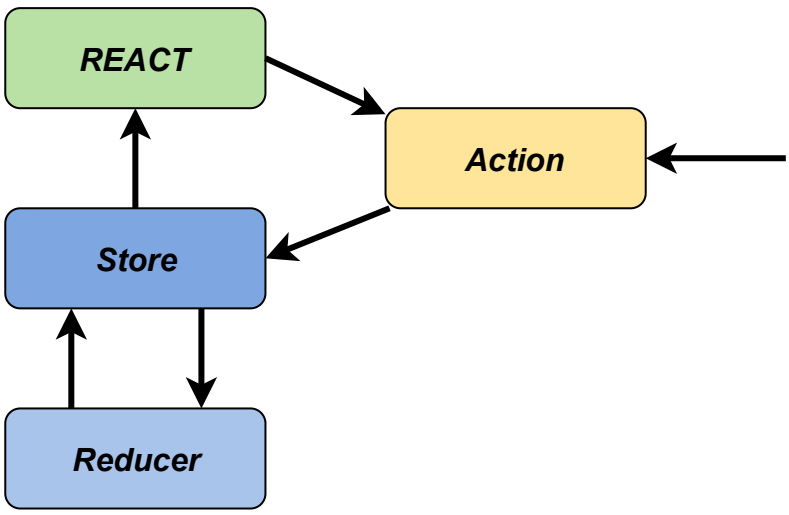
This lesson introduces Redux and its main characteristics

WE'LL COVER THE FOLLOWING

What is it?

Redux is a library that acts as a state container and helps to manage your application data flow. It was introduced back in 2015 at ReactEurope conference (video) by Dan Abramov. It is similar to Flux architecture and has a lot in common with it. In this section we will create a small counter app using Redux alongside React.

Architecture and Main Characteristics



Redux Architecture

Similar to Flux architecture we have the view components (React) dispatching an action. Same action may be dispatched by another part of our system. Like a bootstrap logic for example. This action is dispatched not to a central hub but directly to the store. We are saying “store” not “stores” because there is

only one in Redux. That is one of the big differences between Flux and Redux.

The logic that decided how our data changes lives in pure functions called reducers. Once the store receives an action it asks the reducers about the new version of the state by sending the current state and the given action. Then in immutable fashion the reducer needs to return the new state. The store continues from there and updates its internal state. As a final step, the React component wired to the store gets re-rendered

The concept is pretty linear and again follows the one-direction data flow.

Let's talk about all these pieces and introduce a couple of new terms that support the work of the Redux pattern.