# Introduction to Redux (Action, Reducers and Store)

Redux is a state managing library and an open-source JavaScript library used in JavaScript apps. It simply manages the state of your application or in other words, it is used to manage the data of the application.

Redux was first introduced in 2015 by Dan Abramov and Andrew Clark in 2015. Redux was inspired by Flux but it omitted the unnecessary complexity: it does not have Dispatcher concept, has a single Store and the Action objects is received and handled directly by Store in the Redux.

/\*

**🔁 Dispatcher is a concept present in Flux, not in Redux.**

**💡 In detail:**

**📦 Flux**

* Developed by Facebook.
* **Uses a central Dispatcher** to manage all data flow.
* The dispatcher is like a central hub that **receives actions** and dispatches them to **stores**.
* Architecture: **Action → Dispatcher → Store → View**

**🌀 Redux**

* Inspired by Flux but **simplified the architecture**.
* **Does NOT use a dispatcher**.
* Instead, it has a **single store** and uses **pure reducer functions** to handle state changes.
* Architecture: **Action → Reducer → Store → View**

So in short:  
✅ **Flux → Yes, has Dispatcher**  
❌ **Redux → No Dispatcher**

\*/

Redux is used by ReactJS for building the user interface and to manage the application state.

There are multiple definitions in the market but in simple words we can say Redux is used to manage the data at application level in store. Redux will help us to consume and update the store data from any component.

***Redux Installation:***

To install Redux, React 16.8.3 or higher version is required and is a must.

**Command:**To use Redux with React application.

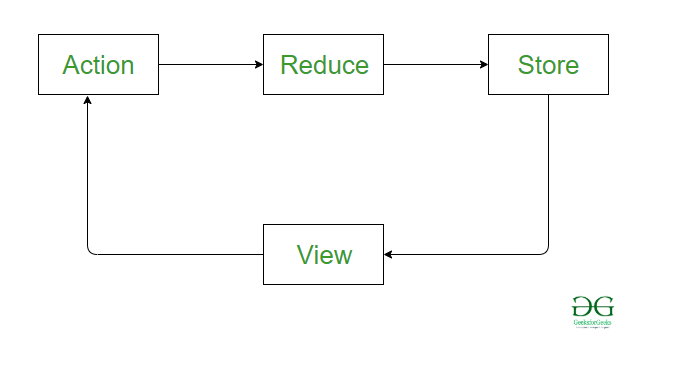
$npm install redux react-redux –save

**Building Parts of redux | *Redux Architecture |* How Redux Works**

As said, Redux allows you to have one global store that handles the state of your application. And every component can have access to that store.

Three building blocks make this ‘magic’ work:

* The store
* The actions
* The reducers



## Store

The store holds your application’s state. Like a cloud out there, or the sun, that each person (component) can see (have access to).

**Actions:** Actions are a plain JavaScript object that contains information. Actions are the only source of information for the store.

Actions have a type field that tells what kind of action to perform and all other fields contain information or data.

And there is one other term called **Action Creators**, these are the function that creates actions. So actions are the information (Objects) and action creator are functions that return these actions.

**Reducers:** As we already know, actions only tell what to do, but they don’t tell how to do, so reducers are the pure functions that take the current state and action and return the new state and tell the store how to do. Reducers **are functions that update your store**, depending on the actions that were sent. When you use the dispatch function to dispatch an action, you are sending the information to the reducer.

The reducer will take that information, and do something, depending on the type property of that action.

Refer :

Anil sidhu

https://www.cluemediator.com/how-to-implement-redux-in-react-js

https://upmostly.com/tutorials/how-to-use-redux-in-react