Assesment-3

design custom flux/redux-pattern to handle state for UI application

Note : don’t use ready redux library

with Three Principles

1.Single source of truth

- The state of your whole application is stored in an object tree within a single store.

2. State is read-only

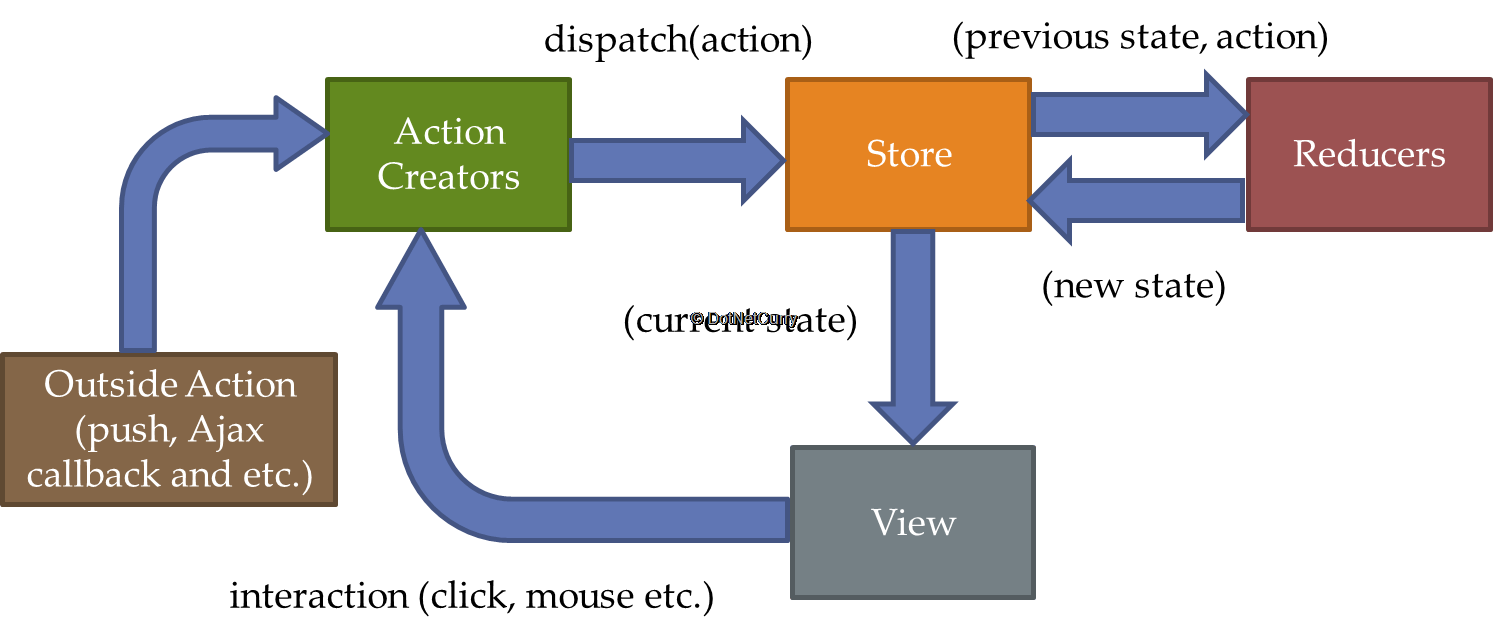
- The only way to change the state is to emit an action, an object describing what happened.

3. Changes are made with pure functions

- To specify how the state tree is transformed by actions, you write pure reducers.

steps:

* define 'action' as contants e.g LOAD\_PRODUCTS
* define action creator functions , to create action object
* define reducer(s) to process relevant actions
* combine all reducer(s) as single root-reducer
* create 'store' with root reducer
* dispatch action to store
* on state change, notify subscribers , with that they can load new-state



for further clarity, read redux documentation