Redux

### \*]  What is Redux?

It is one of the most in-demand libraries for front-end development in today’s growing world. It is defined as the predictable state container mainly designed for JavaScript apps and also it is used for managing the entire state of an application. Redux is very small in size and has no dependencies. It builds applications that are easy to deploy in different environments and easy to test. Redux is very small in size and has no dependencies.

1] **Redux change of state?**

For a release of an action, a change in state to an application is applied, this ensures an intent to change the state will be achieved.

Example : -

1. The user clicks a button in the application.
2. A function is called in the form of component
3. So now an action gets dispatched by the relative container.
4. This happens because the prop (which was just called in the container) is tied to an action dispatcher using mapDispatchToProps (in the container).
5. Reducer on capturing the action it intern executes a function and this function returns a new state with specific changes.
6. The state change is known by the container and modifies a specific prop in the component as a result of the mapStateToProps function.

2] **Where can Redux be used?**  
[Redux is majorly used is a combination with reacting](https://www.educba.com/what-is-redux/). it also has the ability to get used with other view libraries too. some of the famous entities like [AngularJS](https://www.educba.com/angularjs-interview-questions-for-experienced/), [Vue.js](https://www.educba.com/vue-js-vs-jquery/), and Meteor. can get combined with Redux easily. This is a key reason for the popularity of Redux in its ecosystem. So many articles, tutorials, middleware, tools, and boilerplates are available.

3] **What is the typical flow of data in a React + Redux app?**

Call-back from UI component dispatches an action with a payload, these dispatched actions are intercepted and received by the reducers. this interception will generate a new application state. from here the actions will be propagated down through a hierarchy of components from Redux store. The below diagram depicts the entity structure of a redux+react setup.

4] What is store in Redux ?

The store holds the application state and supplies the helper methods for accessing the state.  
register listeners and dispatch actions. There is only one Store while using Redux. The store is configured via the createStorefunction. The single store represents the entire state. Reducers return a state via action

5] **Explain Reducers in Redux?**

The state of a store is updated by means of reducer functions. A stable collection of a reducers form a store and each of the stores maintains a separate state associated for itself. To update the array of donors, we should define donor application

The initial state and action are received by the reducers. Based on the action type, it returns a new state for the store. The state maintained by reducers are immutable. The below-given reducer it holds the current state and action as an argument for it and then returns the next

6] **Redux workflow features?**

Reset: Allow to reset the state of the store  
Revert: Roll back to the last committed state  
Sweep: All disabled actions that you might have fired by mistake will be removed  
Commit: makes the current state the initial state

7] **Explain action’s in Redux?**

Actions in Redux are functions which return an action object. The action type and the action data are packed in the action object. which also allows a donor to be added to the system. Actions send data between the store and application. All information’s retrieved by the store are produced by the actions.