**What is Redux? Why we use this**

Redux is an open-source, JavaScript library used to manage the application state. React uses Redux to build the user interface. It is a predictable state container for JavaScript applications and is used for the entire application’s state management.

Redux **itself is a standalone library that can be used with any UI layer or framework**, including React, Angular, Vue, Ember, and vanilla JS.

**What are the three core principle tha redux follow?**

* A single source of truth.
* The read-only nature of the state.
* The reducer principle.

**What is store in redux?**

A store is **an immutable object tree** in Redux. A store is a state container which holds the application's state. Redux can have only a single store in your application. Whenever a store is created in Redux, you need to specify the reducer.

**How action are define in redux?**

Actions are the JavaScript object which have information. Actions provide information for the store. Actions have a type field that identifies what kind of action to perform and all other fields contain information or data.

function fun\_name(parameters) {

return { // action object

type: 'TYPE\_NAME',

task: task

}

}

**Basic data flow of redux?**

* **Store:** Holds the state of the application.
* **Action:** The source information for the store.
* **Reducer:** Specifies how the application's state changes in response to actions sent to the store.

**Can we have multiple stores in redux?**

**it is possible to create multiple distinct Redux stores in a page, but the intended pattern is to have only a single store**. Having a single store enables using the Redux DevTools, makes persisting and rehydrating data simpler, and simplifies the subscription logic

**What are reducers in redux?**

Reducers, as the name suggests, **take in two things: previous state and an action.** **Then they reduce it (read it return) to one entity: the new updated instance of state**. So reducers are basically pure JS functions which take in the previous state and an action and return the newly updated state

**Why state is immutable in redux**

Because React-Redux performs a shallow check on the root state object to determine if its wrapped components need re-rendering or not, it will not be able to detect the state mutation, and so will not trigger a re-rendering

**Why reduce is pure function?**

**Reducers are just pure functions** that take the previous state and an action, and return the next state. Remember to return new state objects, instead of mutating the previous state.

**What diff b/w mapstateprops() and mapdispatchtoProps()?**

mapStateToProps() is a function used to provide the store data to your component. On the other hand, mapDispatchToProps() is used to provide the action creators as props to your component.

**What is middleware redux?**

Redux Middleware **allows you to intercept every action sent to the reducer so you can make changes to the action or cancel the action**. Middleware helps you with logging, error reporting, making asynchronous requests, and a whole lot more.

**What is redux Thunk?**

Redux thunk acts as middleware which allows an individual to write action creators that return functions instead of actions. This is also used as a delay function in order to delay the dispatch of action if a certain condition is met. The two store methods getState() and dispatch() are provided as parameters to the inner function.

**What diff b/w redux-saga and redux-thunk?**

Redux Saga. **In thunk, action creator does not return an object, it returns a function, In the saga, it allows you to send action normally**. but it has a watcher. whenever a particular action gets dispatched, the watcher catches it.

**How to add multiple middleware to redux?**

Import {createStore, applyMiddleware} from redux;

Const CreateStoreWithMiddleware = applyMiddleware(ReduxThunk, logger)(createStore)

**How to access redux store outside a reach component?**

Import {createStore} from “redux”;

Const store = createStore(myReducer);

Export default store;

**How to use connect from react redux?**

Import {connect} from ‘react-redux’



### ****37. What is the Flux?****

* Flux is the application architecture that Facebook uses for building web applications. It is a method of handling complex data inside a client-side application and manages how data flows in a React application.



* There is a single source of data (the store) and triggering certain actions is the only way to update them.The actions call the dispatcher, and then the store is triggered and updated with their own data accordingly.



* When a dispatch has been triggered, and the store updates, it will emit a change event that the views can rerender accordingly.



### ****38. How is Redux different from Flux?****

|  |  |  |
| --- | --- | --- |
| SN | Redux | Flux |
| 1. | Redux is an open-source JavaScript library used to manage application State | Flux is an architecture and not a framework or library |
| 2. | Store’s state is immutable | Store’s state is mutable |
| 3. | Can only have a single-store | Can have multiple stores |
| 4. | Uses the concept of reducer | Uses the concept of the dispatcher |