1. What is redux ?

Redux is an open-source JavaScript library commonly used with frameworks like React for managing application state. It provides a predictable state container that helps manage the state of an application in a consistent manner. Redux is often used in conjunction with React, although it can be used with other frameworks or even with vanilla JavaScript.

Redux works on the principle of having a single immutable state tree. Changes to the state are made by dispatching actions, which are plain JavaScript objects containing information about what happened. These actions are then processed by special functions called reducers, which specify how the state should change in response to the action.

The key concepts in Redux are:

**Store**: The store holds the entire state tree of the application. It is a single JavaScript object that represents the current state.

**Actions**: Actions are payloads of information that send data from your application to the Redux store. They are plain JavaScript objects and must have a **type** property that indicates the type of action being performed. Additional data can be included as needed.

**Reducers**: Reducers specify how the application's state changes in response to actions. They are pure functions that take the current state and an action as arguments and return a new state. Reducers must be pure functions, meaning they must not mutate the state directly, but instead return a new state object.

**Dispatch**: Dispatch is a method of the Redux store that is used to dispatch actions. When an action is dispatched, Redux calls the appropriate reducer function(s) to update the state.

**Selectors**: Selectors are functions that extract specific pieces of data from the Redux store. They provide a convenient way to access and compute derived state from the store.

1. what is redux thunk used for ?

Redux Thunk is a middleware for Redux, a popular library for managing the state of JavaScript applications. Redux Thunk is used primarily for handling asynchronous actions in Redux.

In Redux, actions are typically plain JavaScript objects describing changes in the state. These actions are dispatched to reducers, which then calculate the new state based on the action type and payload.

However, in many real-world applications, actions may need to perform asynchronous operations, such as fetching data from a server or interacting with a web API. Redux Thunk allows action creators to return functions instead of plain action objects. These functions can have side effects, such as making AJAX requests, and can dispatch actions asynchronously once the operation is complete.

By using Redux Thunk, developers can maintain a clear and consistent approach to managing asynchronous behavior within Redux applications. This middleware simplifies the process of handling asynchronous actions while still maintaining the predictable state management that Redux provides.

1. What is pure component ? when to use pure component over compenent ?

**Pure Component: Implements a shallow comparison in and only re-renders when there are changes in its state or props.**

1. Performance Optimization: Pure components are specifically designed for performance optimization by minimizing unnecessary renders through automatic prop comparison. ...
2. Shallow Comparison: ...
3. Usage with Hooks:
4. What is the second argument aotionally be passed tosetstate and what is its pur pose

The second parameter to setState() is an optional callback function that will be executed once setState is completed and the component is re-rendered. Generally we recommend using componentDidUpdate() for such logic instead.