React JS

React Redux

Agenda

01 What is Redux?

02 Why use Redux?

03 Principles of Redux

04 Redux Concepts

05 React - Redux

06 Connect Function

What is Redux?

What is Redux?

As we have already discussed, React supports two kinds of state, Local State and Global State

What is Redux?

In React there are two ways of dealing with global state: Lifting Up State Pattern and Context. These are great for small to mid sized applications

What is Redux?

But, In an application that has a lot of state that needs to be shared and manipulated by multiple component, these solutions really complicate the code and are even difficult to implement

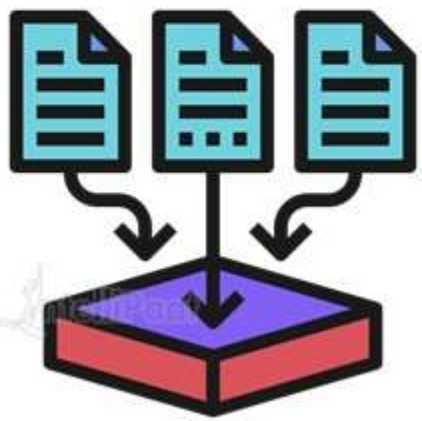
What is Redux?

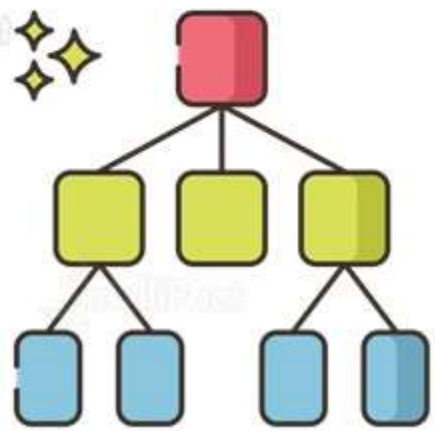
Redux is a JavaScript library that solves this problem. It maintains the state of an entire application in a single immutable state tree (JS object), which can't be changed directly

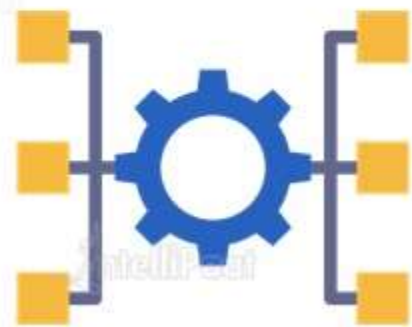
Why use Redux?

Why use Redux?

ReactJS allows us to build our applications user interface by building components that manage their own state internally without any external tool or library

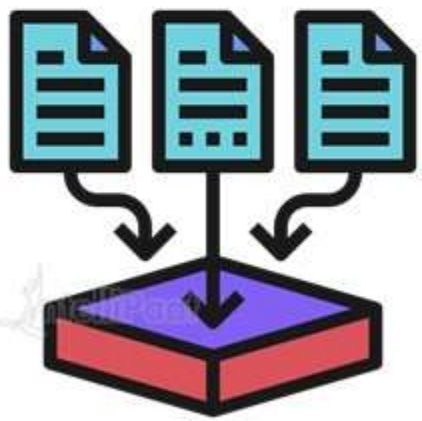


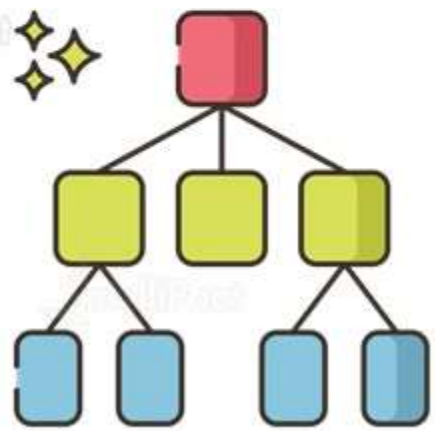


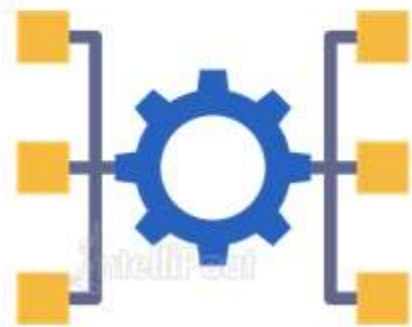


Why use Redux?

In an application with fewer components this approach works really well and allows developers to build small isolated user interface components

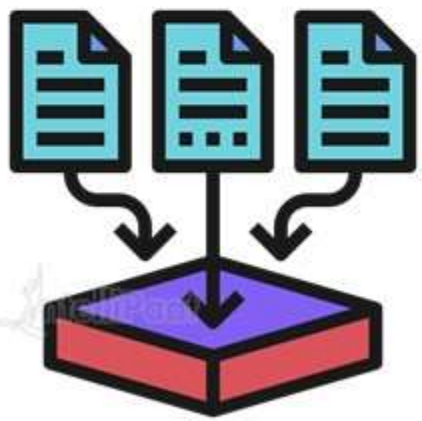


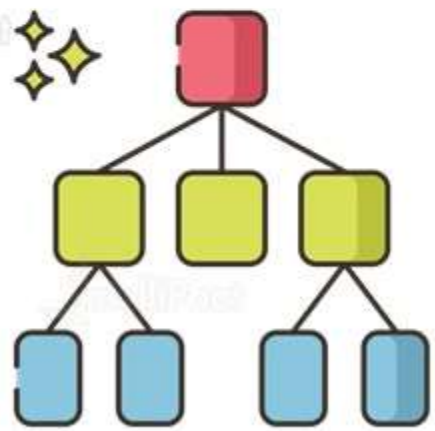


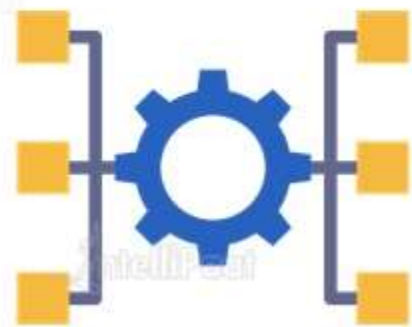


Why use Redux?

But as the application size grows managing states shared across components becomes a really difficult task.

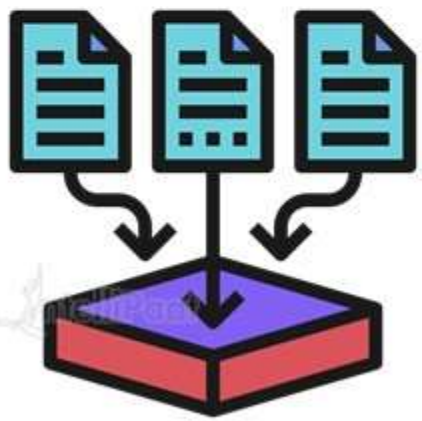


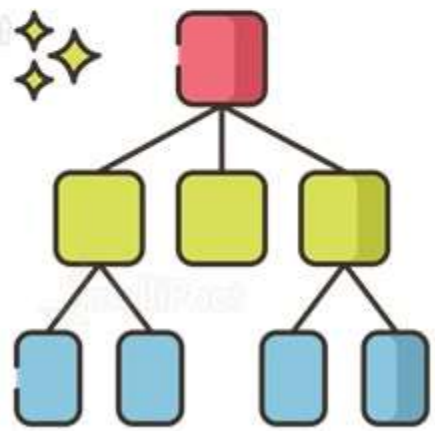


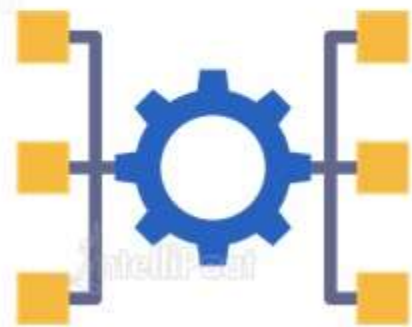


Why use Redux?

As data needs to be shared among components, it might be confusing to actually know where a state should live

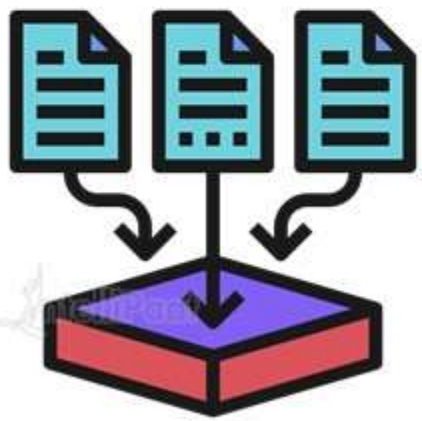


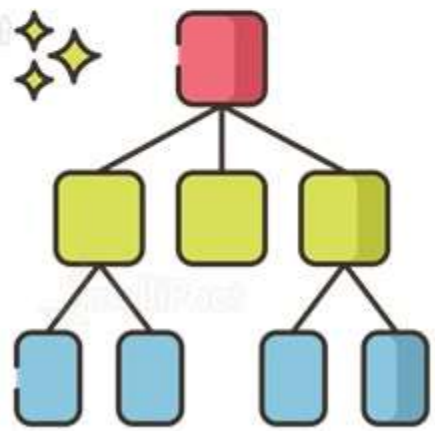


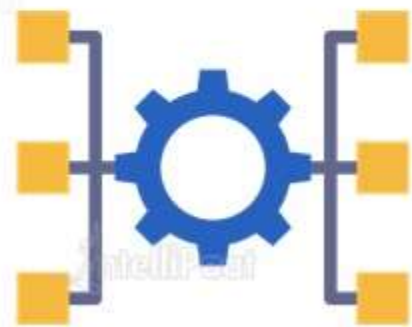


Why use Redux?

Since state management gets messy as the app gets complex, we need a state management tool like Redux that makes it easier to maintain these states across several components in a consistent manner







Principles of Redux

Principles of Redux

There are three fundamental principles that Redux follows

Single source of truth

State is read-only

Changes are made with pure functions

It means that there is only one place which represents global state of your application and our application UI will re-render when this state changes, this makes your application more predictable

Principles of Redux

There are three fundamental principles that Redux follows

Single source of truth

State is read-only

Changes are made with pure functions

In Redux, we cannot mutate (modify) the state directly, the only way to do that is to emit an action, which is an object describing what needs to happen

Principles of Redux

There are three fundamental principles that Redux follows

Single source of truth

State is read-only

Changes are made with pure functions

In Redux we use reducers which are pure functions that take the current state and an action, and return the next state. A pure function is a function that Given the same input, will always return the same output and produces no side effects.

Redux Concepts

Redux Concepts

There's some important Redux terms and concepts that we need to be familiar with before we can continue

- Actions
- Action Creators
- Reducers
- Store
- Dispatch
- Selectors

An action is a plain JavaScript object that has a type field. It can thought of as an event that describes something that happened in the application. For example: When a todo is added in a todo list

Redux Concepts

There's some important Redux terms and concepts that we need to be familiar with before we can continue

- Actions
- Action Creators
- Reducers
- Store
- Dispatch
- Selectors

The type field should be a string that gives the action a descriptive name, like "todos/todoAdded". It is usually written created as "domain/eventName“

Redux Concepts

There's some important Redux terms and concepts that we need to be familiar with before we can continue

- Actions
- Action Creators
- Reducers
- Store
- Dispatch
- Selectors

An action object can have other fields with additional information about what happened. By convention, we put that information in a field called payload

Redux Concepts

There's some important Redux terms and concepts that we need to be familiar with before we can continue

- Actions
- Action Creators
- Reducers
- Store
- Dispatch
- Selectors

Some action objects can be a little complicated and creating them by hand all the time can be a little cumbersome and error prone

Redux Concepts

There's some important Redux terms and concepts that we need to be familiar with before we can continue

- Actions
- Action Creators
- Reducers
- Store
- Dispatch
- Selectors

Which is why we use functions to create our actions that take a few inputs as function arguments and return an Action with type and payload

Redux Concepts

There's some important Redux terms and concepts that we need to be familiar with before we can continue

Actions | Action Creators | Reducers | Store

Dispatch | Selectors

These functions are called Action Creator. We typically use these so we don't have to write the action object by hand every time

Redux Concepts

There's some important Redux terms and concepts that we need to be familiar with before we can continue

- Actions
- Action Creators
- Reducers
- Store
- Dispatch
- Selectors

As in Redux state is read only, so to make a changes to state such as adding or deleting items we need to copy the state, manipulate it and set the manipulated state as new state

Redux Concepts

There's some important Redux terms and concepts that we need to be familiar with before we can continue

Actions | Action Creators | Reducers | Store

Dispatch | Selectors

Action objects are used to describe the change to the state, but these changes are actually performed in a function called reducer

Redux Concepts

There's some important Redux terms and concepts that we need to be familiar with before we can continue

Actions

Action Creators

Reducers

Store

Dispatch

Selectors

A Reducer function simple takes the current state and action as argument, copies the state, makes changes to the copy based on the action and returns the new state

Redux Concepts

There's some important Redux terms and concepts that we need to be familiar with before we can continue

- Actions
- Action Creators
- Reducers
- Store
- Dispatch
- Selectors

In Redux a Store is the object that sores the entire global state of an application that uses Redux

Redux Concepts

There's some important Redux terms and concepts that we need to be familiar with before we can continue

Actions

Action Creators

Reducers

Store

Dispatch

Selectors

A Store is created by passing in a single or multiple reducers and has a method called getState that returns the current state value

Redux Concepts

There's some important Redux terms and concepts that we need to be familiar with before we can continue

Actions

Action Creators

Reducers

Store

Dispatch

Selectors

As mentioned before Store is the place where all the global state is stored and is the single source of truth in a Redux Application

Redux Concepts

There's some important Redux terms and concepts that we need to be familiar with before we can continue

Actions | Action Creators | Reducers | Store

Dispatch | Selectors

A Store in Redux has a method on it called dispatch, which is called with an action object when we wish to update the current state

Redux Concepts

There's some important Redux terms and concepts that we need to be familiar with before we can continue

Actions

Action Creators

Reducers

Store

Dispatch

Selectors

When the dispatch method is called with an action the sore runs the reducer function with current state and action and updates the current state to be the state returned by the reducer

Redux Concepts

There's some important Redux terms and concepts that we need to be familiar with before we can continue

Actions | Action Creators | Reducers | Store

Dispatch | Selectors

Think of dispatching actions as "triggering an event" in the application. Something happened, and we want the store to know about it.

Redux Concepts

There's some important Redux terms and concepts that we need to be familiar with before we can continue

- Actions
- Action Creators
- Reducers
- Store
- Dispatch
- Selectors

As the state in our application grows, it can become increasingly difficult to extract just the information you want out of it

Redux Concepts

There's some important Redux terms and concepts that we need to be familiar with before we can continue

Actions | Action Creators | Reducers | Store

Dispatch | Selectors

To deal with this we use Selectors. These are just plain functions that receive the state as an argument and extract specific pieces of information from the current state

Redux Concepts

There's some important Redux terms and concepts that we need to be familiar with before we can continue

- Actions
- Action Creators
- Reducers
- Store
- Dispatch
- Selectors

As an application grows bigger, this can help avoid repeating logic as different parts of the app need to read the same data

Hands On: Redux

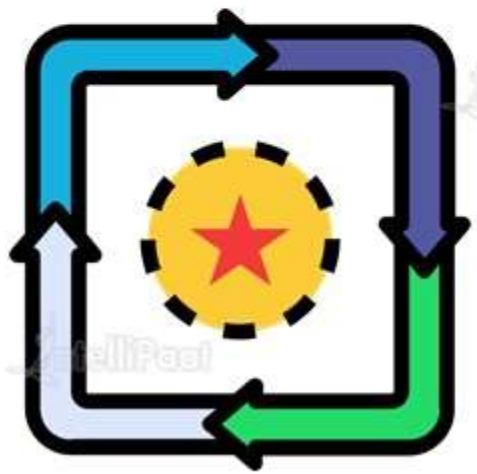
React - Redux

React - Redux

Since Redux is a JavaScript Library for State Management, it can be used with any applications weather it a React App or Angular App etc

React - Redux

For React to work well with Redux we need to use another package called React - Redux



React - Redux

It is a package created and maintained by Redux team, and kept up-to-date with the latest APIs from Redux and React

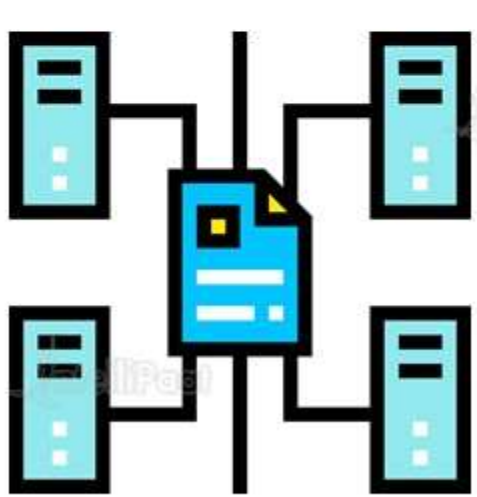
React - Redux

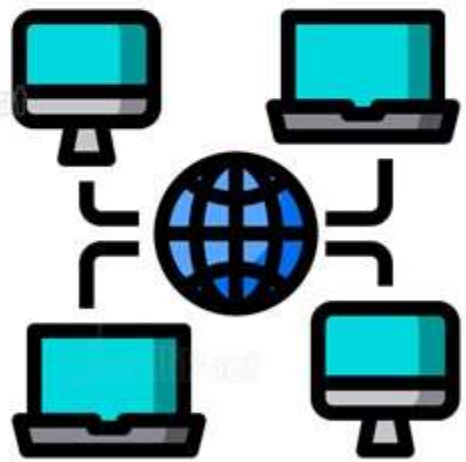
It creates wrapper components that manage the store interaction logic for you, so you don't have to write it yourself

Connect Function

Connect Function

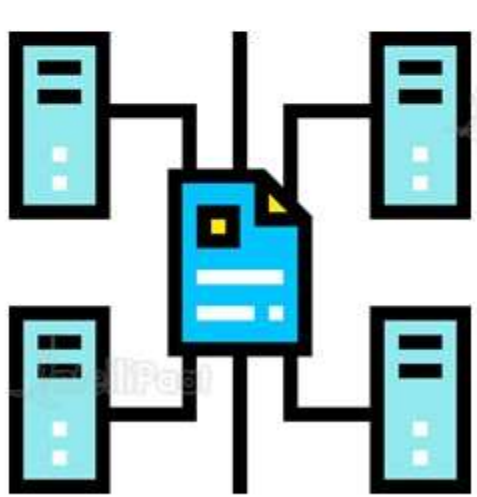
When using React Redux when we wish to connect our Redux Store to React Components we need to the connect() function

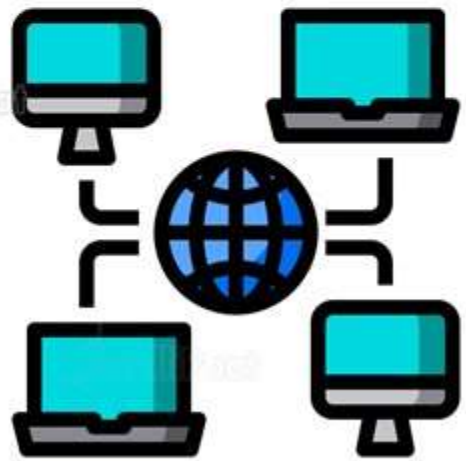




Connect Function

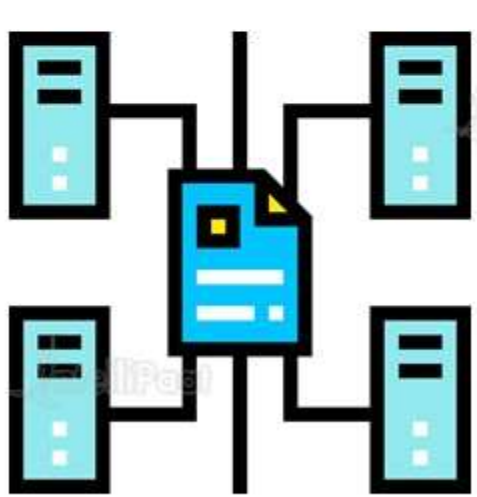
It provides component with the pieces of the data it needs from the store, and the functions it can use to dispatch actions to the store.

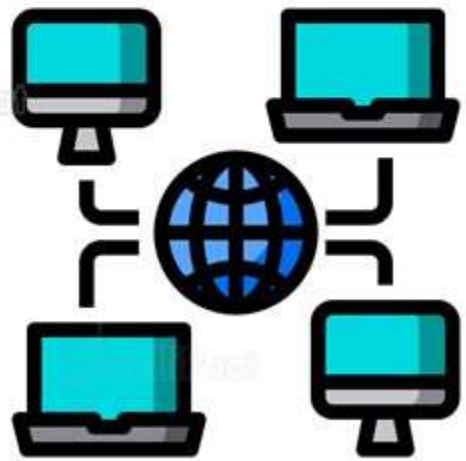




Connect Function

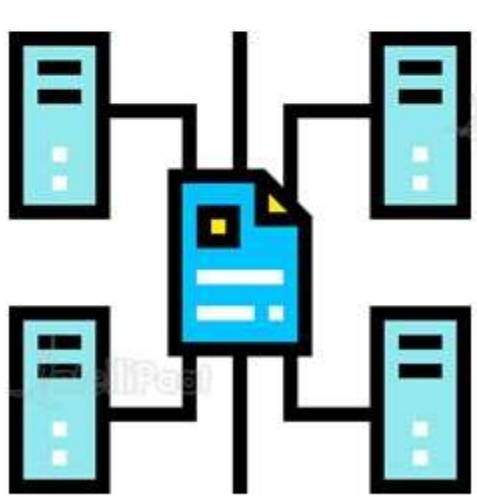
It does these things by accepting two optional functions as its parameters that are named **mapStateToProps** and **mapDispatchToProps** respectively

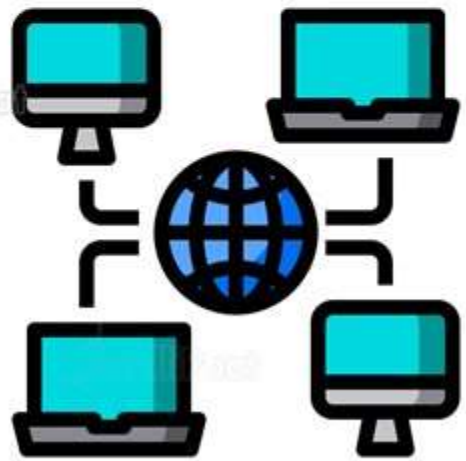




Connect Function

When connect is called with these functions it returns another function which is called by passing a React Component as argument so that it can receive data as props and display it





Demo: React - Redux

India: +91-7847955955

US: 1-800-216-8930 (TOLL FREE)

support@intellipaat.com

24/7 Chat with Our Course Advisor