React Redux BootCamp

About This Course

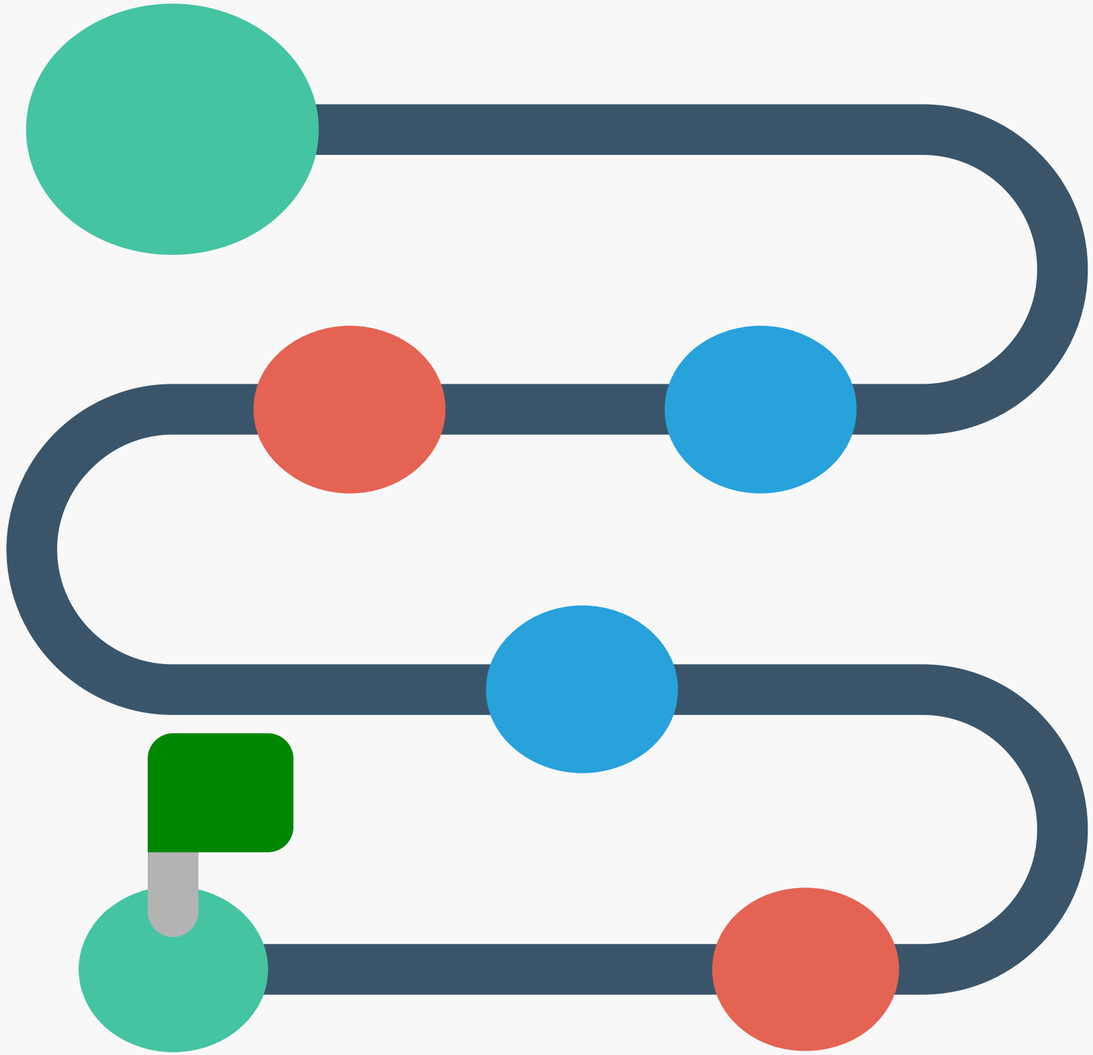
What is Redux ?

What is Redux ?

- Redux is an open-source JavaScript library for managing and centralizing application state

- A Predictable State Container for JS Apps

Redux Roadmap



Redux Roadmap

1 REDUX

2 REACT REDUX

3 REDUX TOOLKIT

4 RTK QUERY

What is state?

What is state?

👉 Any data in your application that can change based on conditions.

👉 It's a data store that is used to manage the component data

What is State Management ?

👉 It's the ability to control the information that is passed between React components.

👉 State management is the process of determining how to manage state information in a web application.

👉 State management can be used to track

- User data
- Session data
- Application data
- Component data

Types of Data Source

- Server Data
- Global State
- Local State
- URL state
- Routing
- Local Storage

Local State Management

Local State — Local state refers to the data that we manage in a particular component.

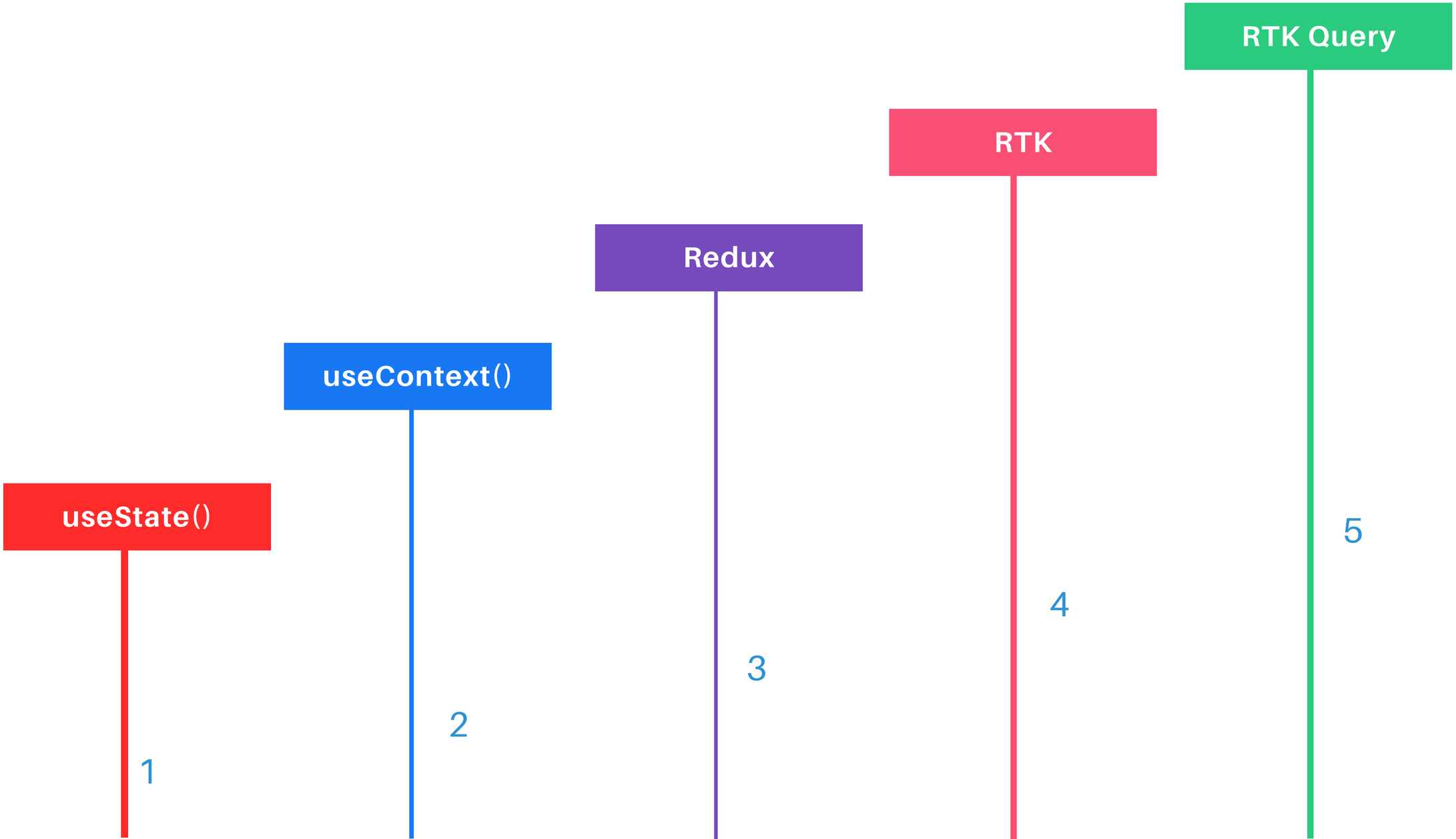
Global State — data we manage across multiple components.

URL State — Data that exists on our URLs, including the pathname and query parameters.

localStorage State — Data from browser API localstorage

Server State — Data that comes from an external server

Ways of Managing State in react


useState()
useContext()
Redux
RTK
RTK Query
1
2
3
4
5

State Management Libraries

State Management Libraries

pmndrs/zustand: 🐻 Bear necessities for state management in React

🐻 Bear necessities for state management in React. Contribute to pmndrs/zustand development by creating an account on GitHub.

GitHub

Recoil

A state management library for React

Recoil

A state management library for React.

recoiljs.org

Jōtai

Primitive and flexible state management for React

Jotai, primitive and flexible state management for React

Jotai takes a bottom-up approach to React state management with an atomic model inspired by Recoil. One can build state by combining atoms and renders are optimized based on atom dependency. This...

Jotai

Redux

Redux - A predictable state container for JavaScript apps. | Redux

A predictable state container for JavaScript apps.

redux.js.org

Getting Started With Redux

Redux Basics

When To Use Redux?

When To Use Redux ?

- 👉 You should use Redux when you have a complex state object that is difficult to manage with the local state alone
- 👉 The app state is updated frequently
- 👉 The logic to update that state may be complex
- 👉 The app has a medium or large-sized codebase, and might be worked on by many people

Redux Benefits

Redux Benefits

- 👉 Predictable
- 👉 Centralized state management
- 👉 Debuggable

Redux Terminologies

Redux Terminologies

Actions

Actions are the driving force of every dynamic application, as they are the medium by which all changes are communicated within a Redux application

Reducers

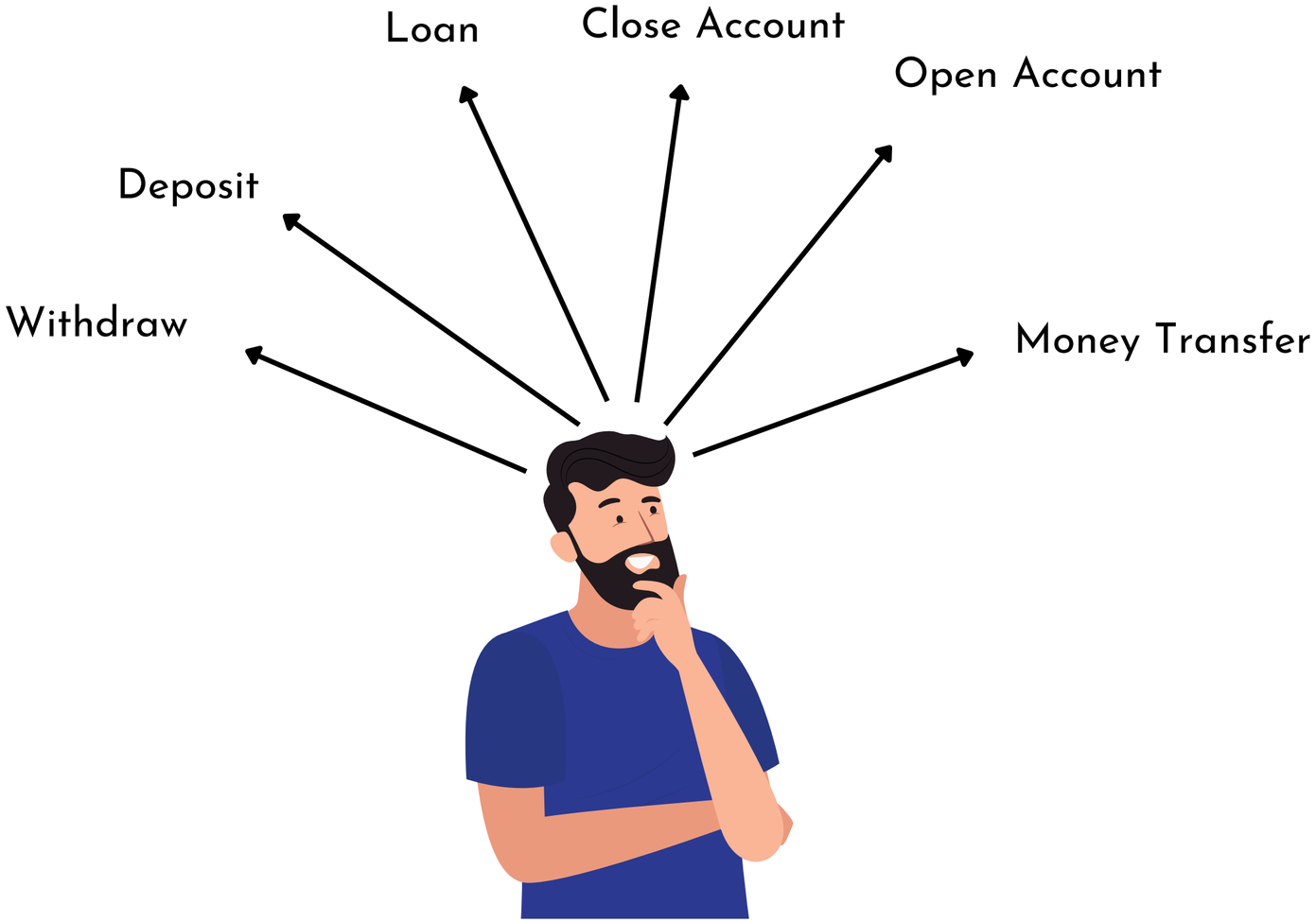
Reducers are event listeners which handle events based on the action type

Store

It stores the application data

How Redux Works

ACTION



Action

HOW REDUX WORKS

Total Balance $5000

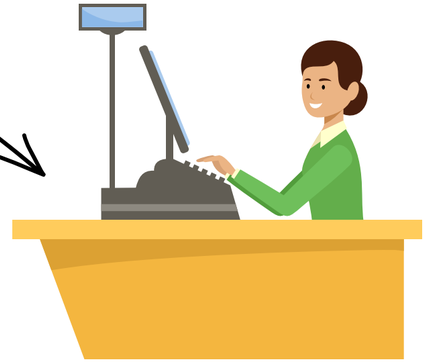
BANK

store

Deposit Action Type - Dispatch

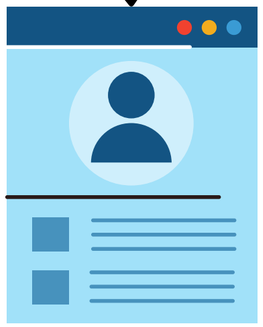
100

Action



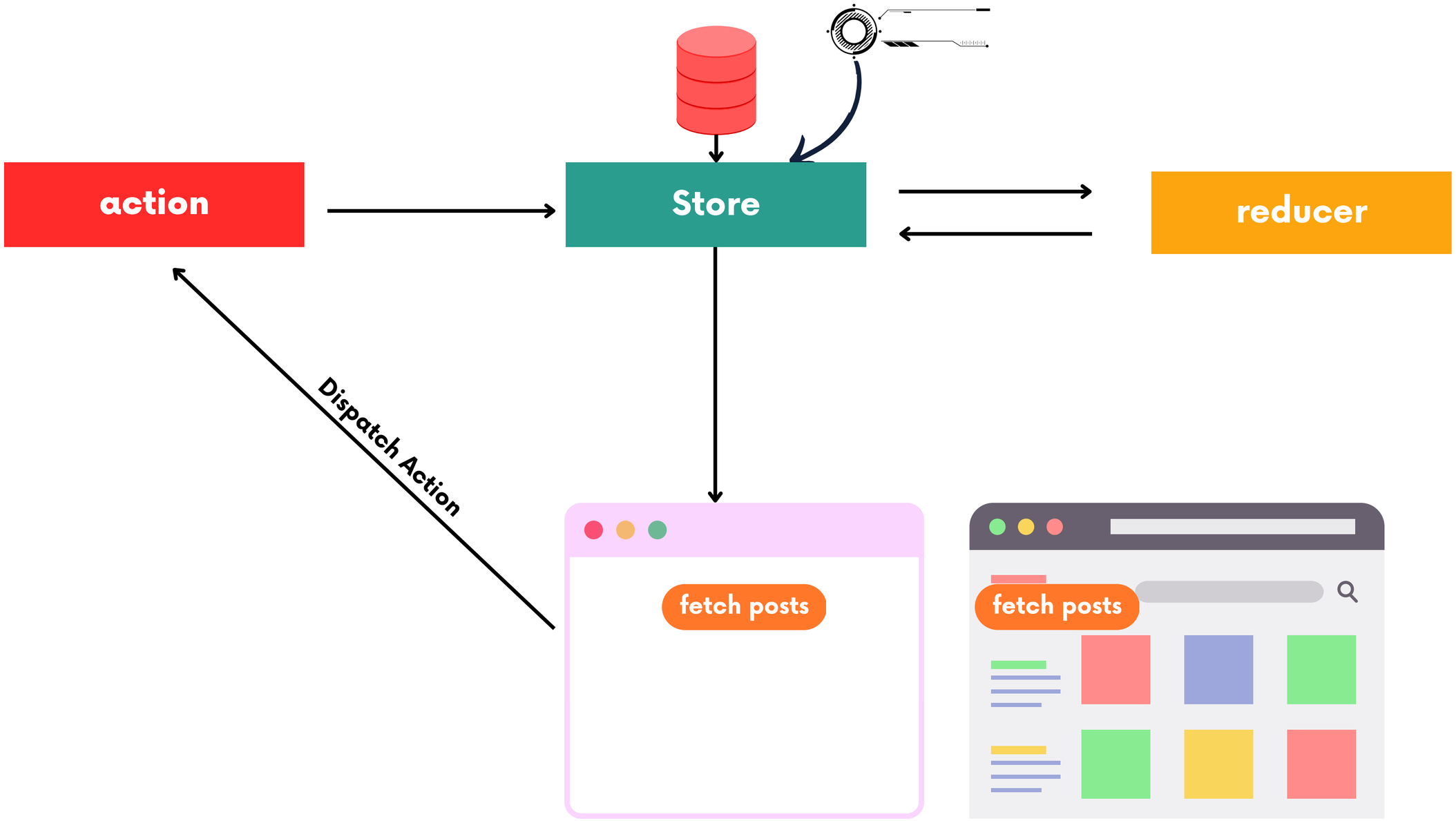
Reducer

Deposit Action Type - Dispatch



Total Balance $6000

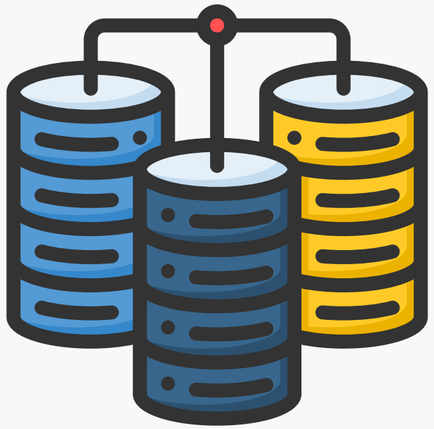
How Redux Works Analysis


action
Store
reducer
Dispatch Action
fetch posts
fetch posts

Assignment

Use any practical senario to explain, how redux works. For example in education sector

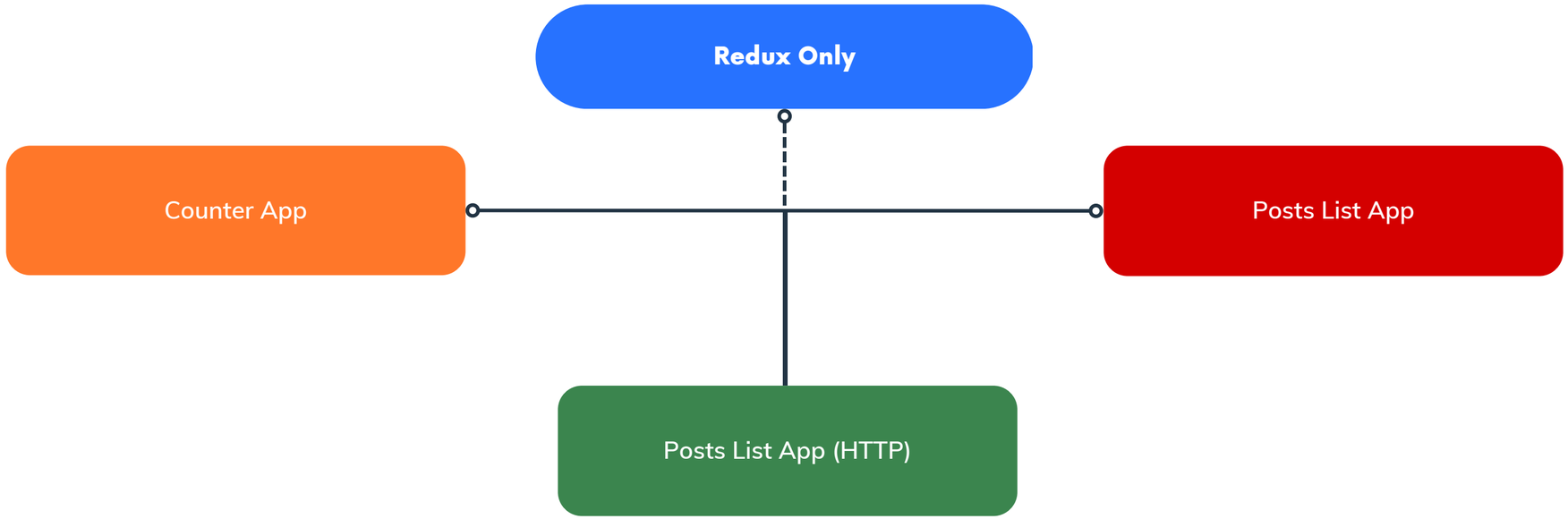
Store Methods



Redux Core Library

Understanding the bare bone of redux is the great way to master react redux

Redux Core Library



Action

vs

Action Creator

Action

An action is simply a JavaScript object that contains information about an event that has occurred in your app

Action Creator

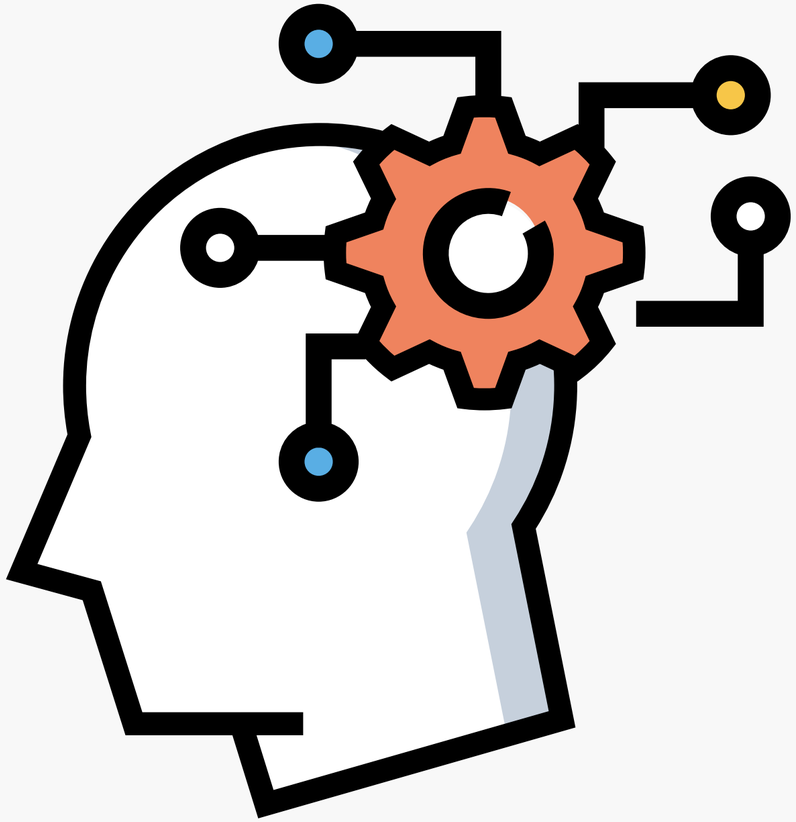
Action creators are functions that create and return actions

Action Properties

Action Properties

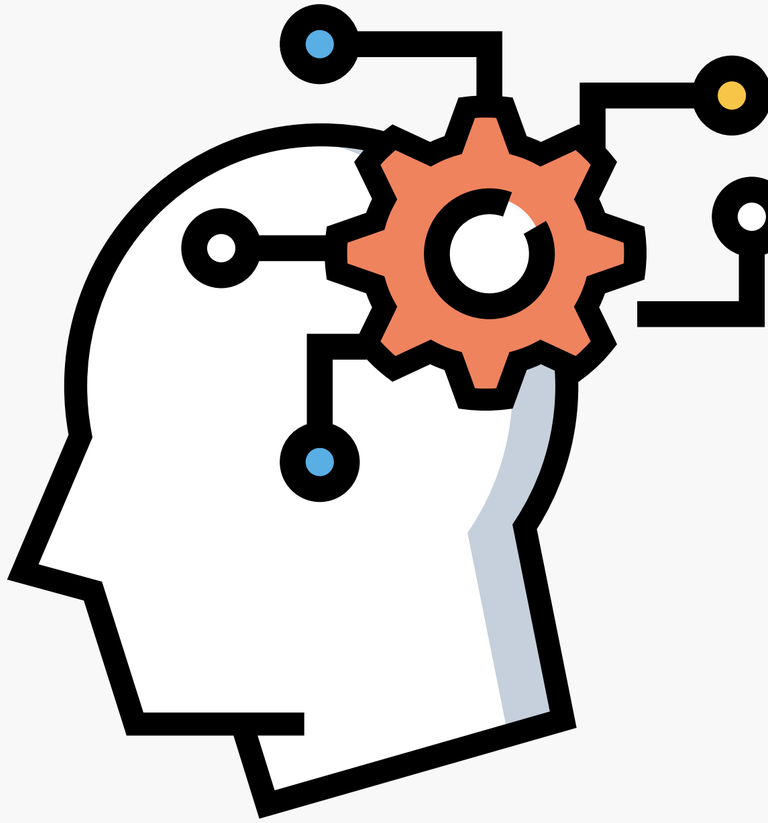
- It's a plain JavaScript object
- It has a type field as a property which is a required
- It's an event that occurs in your application
- It can accept additional properties (payload). This is optional
- A function that returns an action is called an action creator

REDUCER



REDUCER

A reducer is a function that receives the current state and an action object, decides how to update the state base on the action, and returns the new state



RULES OF REDUCER

RULES OF REDUCER

- The new state value should only be calculated based on the state and action arguments.
- Reducers are not allowed to modify the existing state.

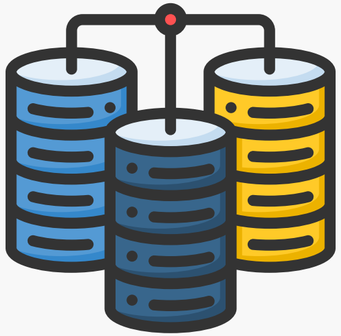
STORE

STORE

When it receives an action that causes a change to the state, the store will notify all the registered listeners that a change to the state has been made. This will allow various parts of the system, like the UI, to update themselves according to the new state

- It stores the application data
- It doesn't contain business logic
- It receives actions and pass to all all the registered middleware
- The only way to change the state inside it is to dispatch an action

Store Methods



getState() → Returns the current state tree of your application

dispatch() → This is the only way to trigger a state change.

subsribe() → Listening to any change

Combine Reducer

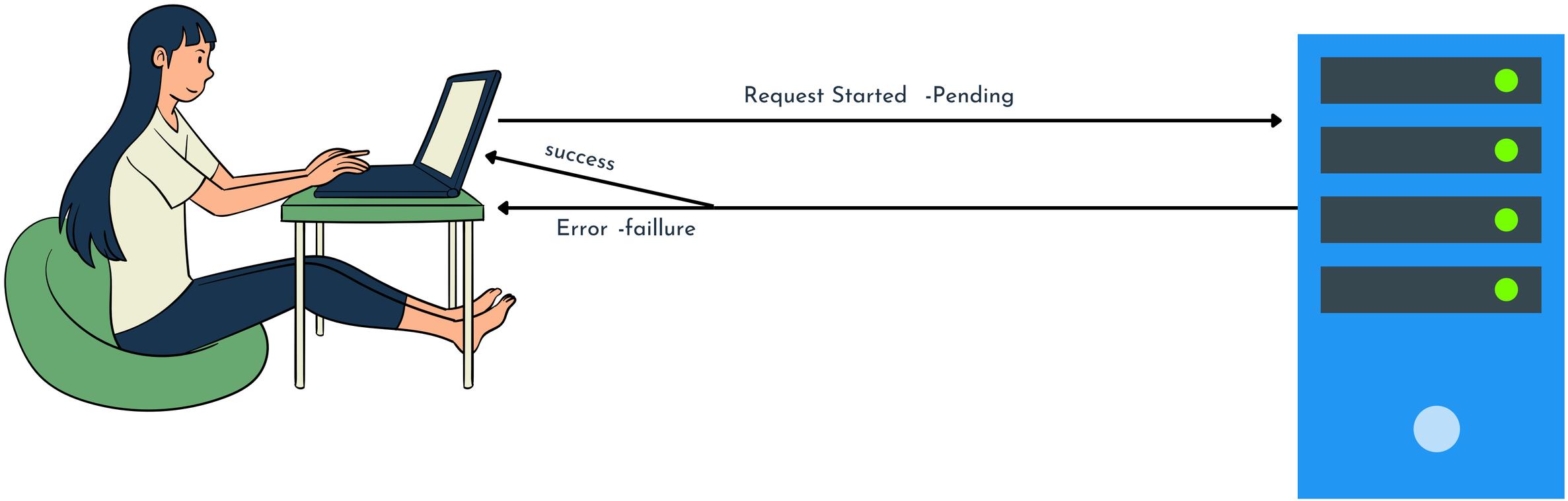
It's a function that combine individual reducers to pass to redux store

1. INITIAL STATE
2. ACTIONS
3. REDUCERS
4. COMBINE REDUCERS
5. STORE
6. STORE ENHANCEMENT
7. MIDDLEWARES

Asynchronous Redux

Fetching data from external server - (API)

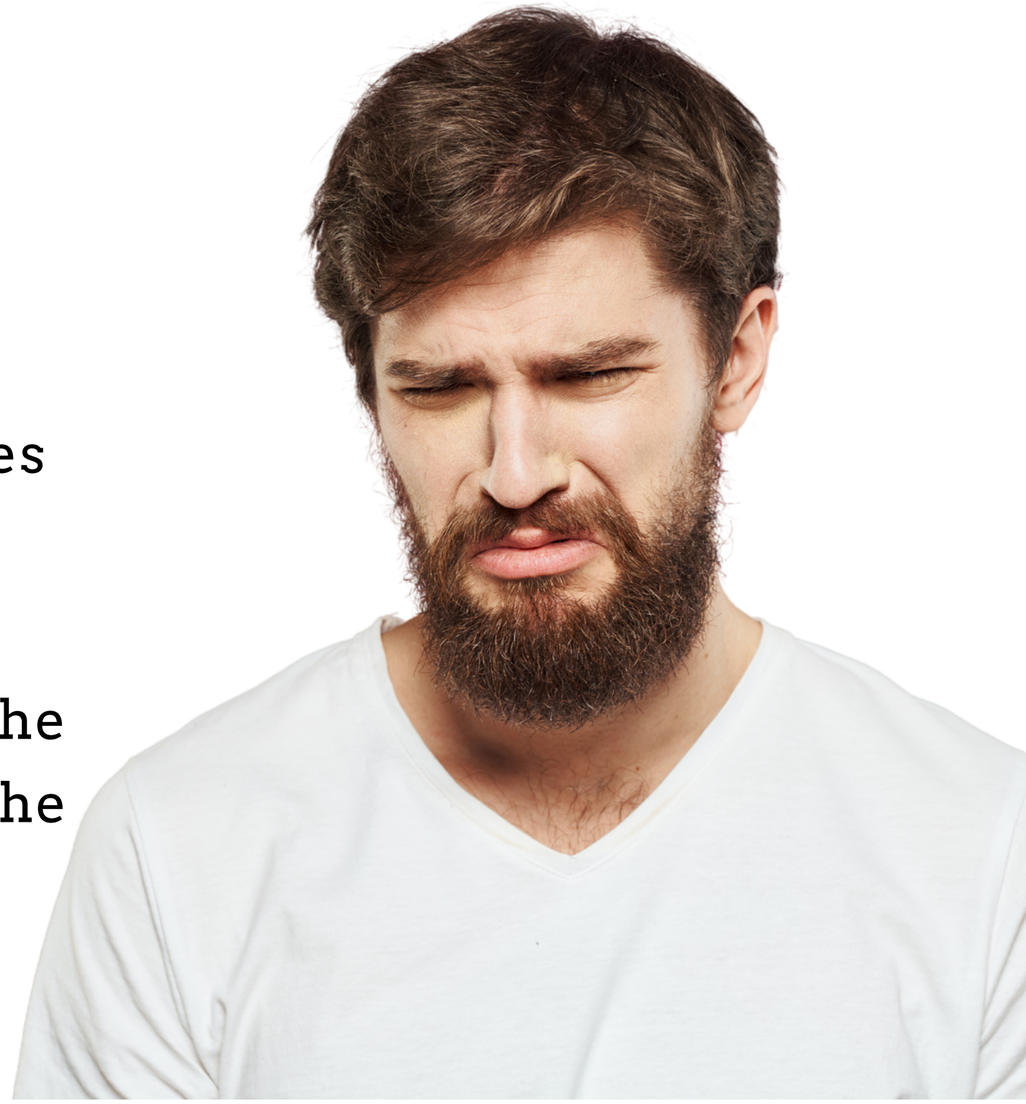
PROJECT


Request Started -Pending
success
Error -faillure

Side effects of Making API Call without handling promise in redux

Asynchronous Redux

- data is never received, or is received out of order.
- it can make it difficult to debug your code
- the redux store will not be updated and no changes will be made to your application state
- Reducers are imediatelly return a new data if the correct action is dispatch without waiting for the action payload



Store Configuration

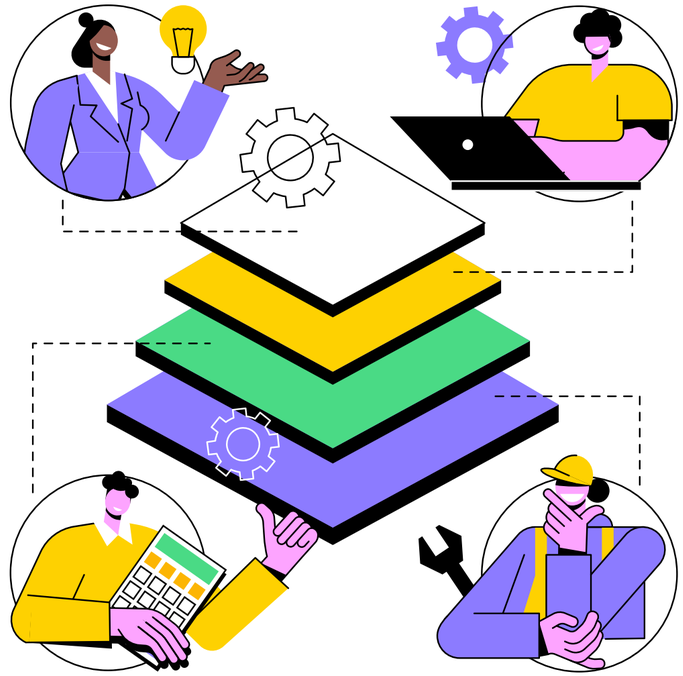
Advanced Store setup

Middleware

Redux middleware is a powerful tool that can be used to customize and extend the functionality of Redux

Middleware

Middleware is basically a function that takes in an action and can decide how to handle it



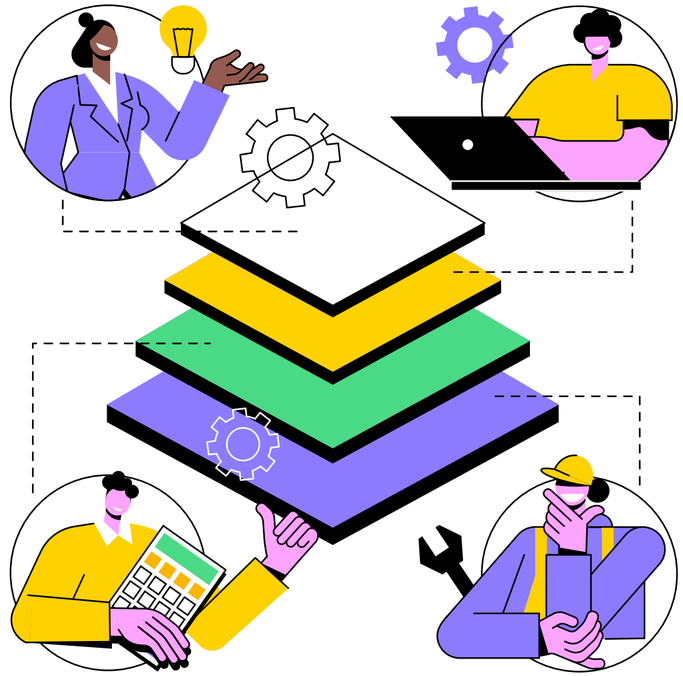
Types Of Middlewares

Types Of Middlewares

THIRD PARTY MIDDLEWARS

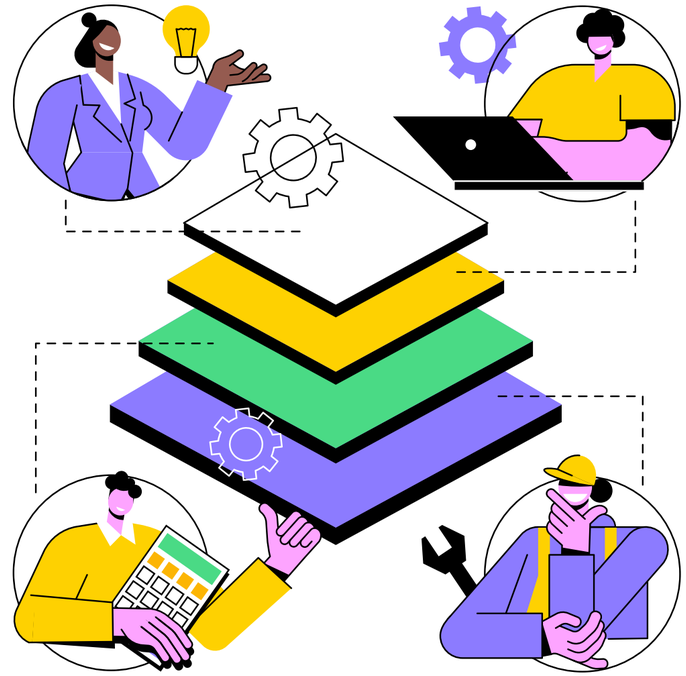
CUSTOM MIDDLEWARES

Uses Of Middleware



Uses Of Middleware

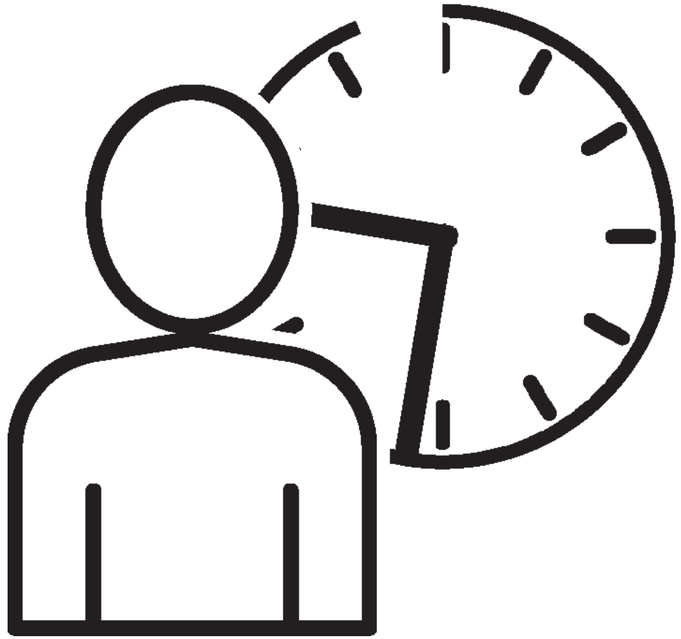
- Handle the action
- Dispatch a new action (i.e. create a side-effect such as making an API call)
- Log the action to the console / inside the browswer by using redux-dev-extension tool



Middlewares are used to enable advanced functionality in a Redux store that would not be possible with just a reducer alone.

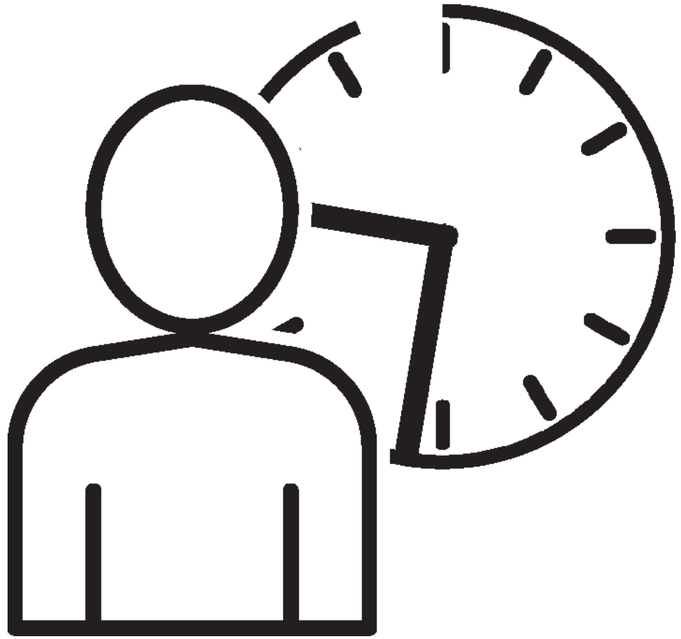
Middlewares are composed and executed within a Redux store using the applyMiddleware() function.

Redux Thunk



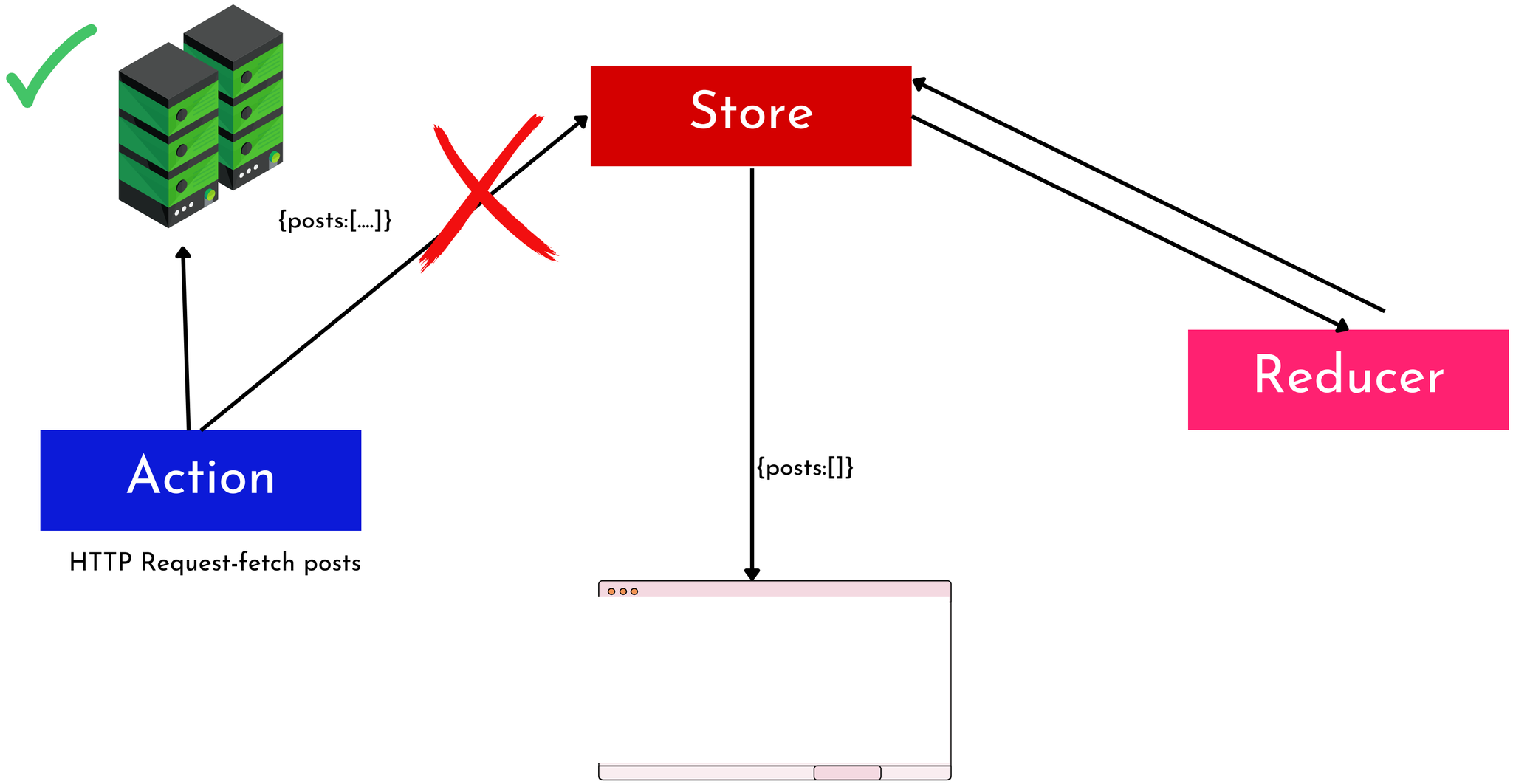
Redux Thunk

Redux Thunk is a middleware that allows you to write asynchronous actions.



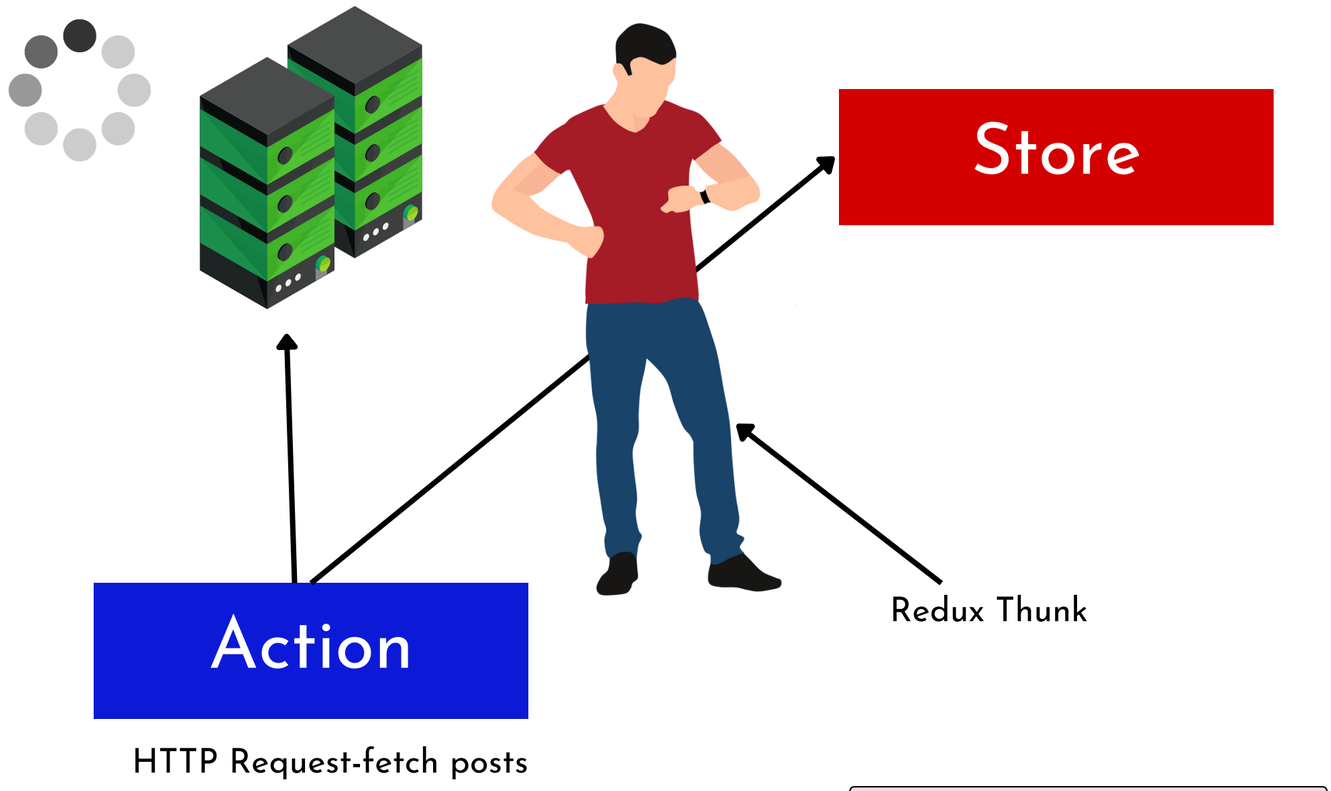
Without Redux Thunk

Without Redux Thunk

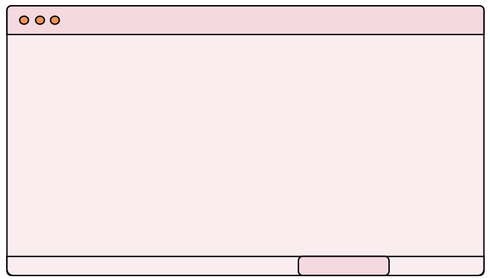


With Redux Thunk

Without Redux Thunk



Reducer



Facts About Redux Thunk

Facts About Redux Thunk

It's a function (action creator) that return a function instead of an action object.

This function receives the dispatch method as an argument, which allows you to dispatch actions inside the function.

This is often used when you need to perform an async operation, such as making an AJAX request, before dispatching an action.

React Redux

Brainstorming

Brainstorming

What actions to create

Is it Async Actions

What will be the InitialState

How to dispatch the action

How will be the reducer