

Essential Concepts of Redux



Essential Concepts of Redux

© IBM Corporation. All rights reserved.





What you will learn



Discuss Redux concepts such as action, store, and reducer



Describe the principles behind Redux architecture



Explain the terms "dispatching" and "action creators"



Relate the concepts action, store, and reducer to a user interface example



Why Redux

No need to manage component states

States managed at the application level

Allows for immutable component properties





Redux elements

Store:

All current states in the app

Action:

Object that indicates state change intent

Reducer:

Specifies how to change the state





Shopping cart example



Add to cart







Shopping cart example



Add to cart







Dispatching

Add to cart

Action creator: addItem() function

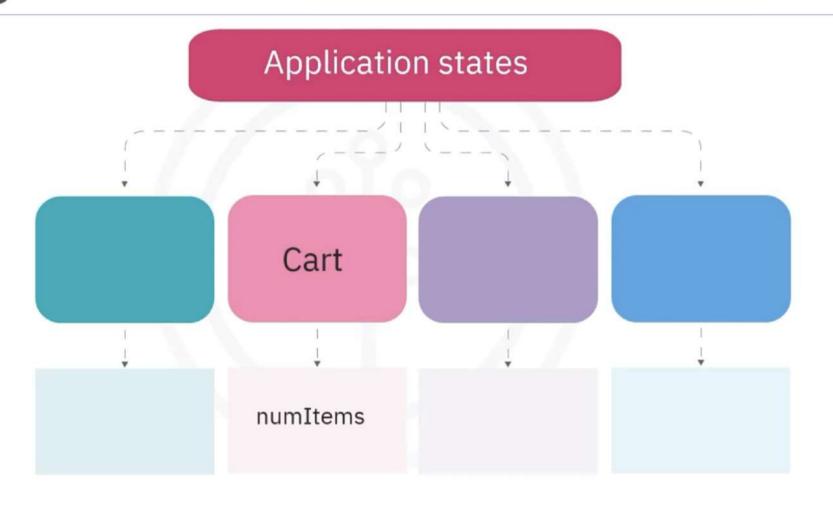


"Dispatching an action"





Store







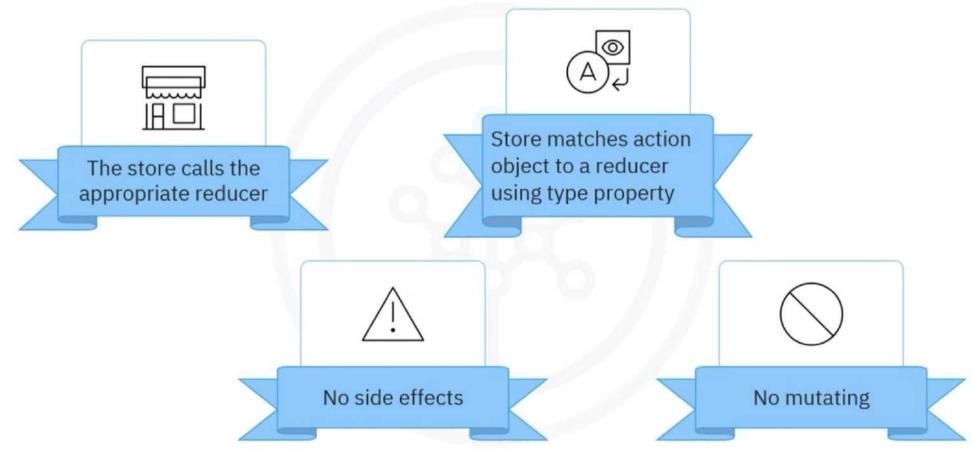
Action object properties

Properties Type Payload Indicates the reducer Other data to update the state





Reducer







Why this architecture?







Recap

In this video, you learned that:

- The action object describes the action that needs to be performed
- The store maintains all of the application's current states
- A reducer tells the store how to change the state
- The dispatching action process entails using an action creator function, which returns the action object and sends the object to the store using a dispatch function
- The Redux architecture makes states predictable because changes are explicit through actions, reducers, and the application store



