

# Introduction to Redux.js

**Relevel**  
by Unacademy



# Topics Covered



- **Introduction**  
What is Redux.js? Where is it used?



- **Components of Redux**  
What elements of Redux help us centralize our state?



- **Redux Form**  
We will be creating a form bootstrapped from redux-forms



- **Practice Questions**  
Assignments

# Introduction to Redux.js

- Redux is a JavaScript library developed by Dan Abramov & Andrew Clark in 2015.
- It is used for containerizing & centralizing the state of the application.
- Predictable-Uses pure functions that improve reliability.
- Flexible in nature-Can be used with any UI layer.



# Why learn Redux.js

Redux isn't an absolute necessity tool for web development, but for state management, it's essential; here is why:

- React components can pass props only from top to bottom(parent to child) order.
- If the change in the state of one component affects the state of others, it becomes challenging to update the states.
- Redux helps us centralize the application's state through a store and provides ways to access/update it via reducers/actions.



# Components of Redux.js

Redux is composed of 4 primary components:

- UI components
- Actions
- Reducers
- Store



# UI Components

Redux is composed of 4 primary components:

- The elements showcased to the end user
- Contain initial state
- All the state updates are ultimately visible on UI level
- Eg: Login page



# Actions

- Objects that are dispatched by the UI Components.

- Syntax:

```
const action =  
  { type: ADD_NAME,  
    data: name }
```

- type is a property that specifies the nature of action to be invoked.
- data is a property that contains actual data specific to action type.
- E.g. Adding a name to a list of names.



# Reducer

- A reducer is a pure function that takes two.

- Syntax:

```
const action =  
  { type: ADD_NAME,  
    data: name }
```

- type is a property that specifies the nature of action to be invoked.
- data is a property that contains actual data specific to action type.
- E.g. Adding a name to a list of names.





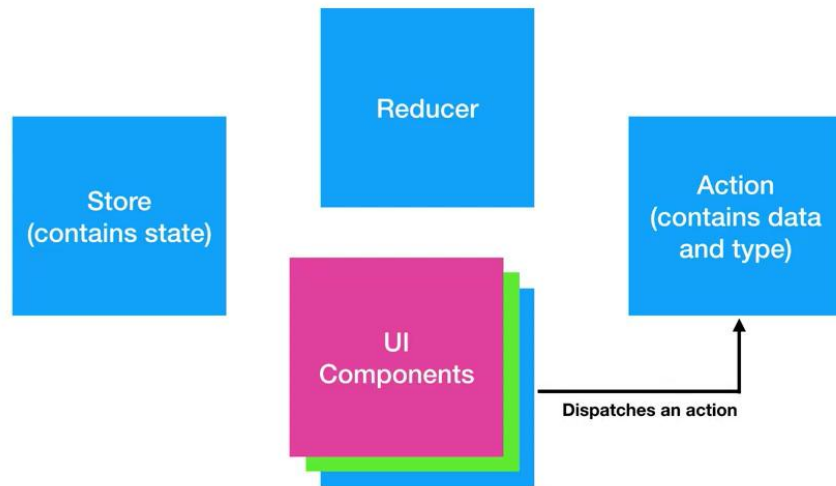
# Reducer

- A store is a warehouse of application states.
- It contains initial state of all components and watches for changes from reducer.



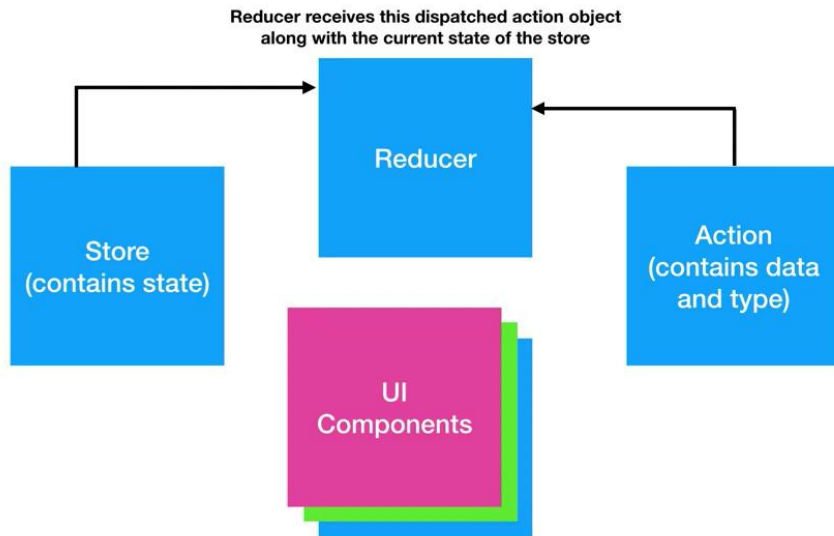
# Data flow in Redux

- The UI components dispatch actions.



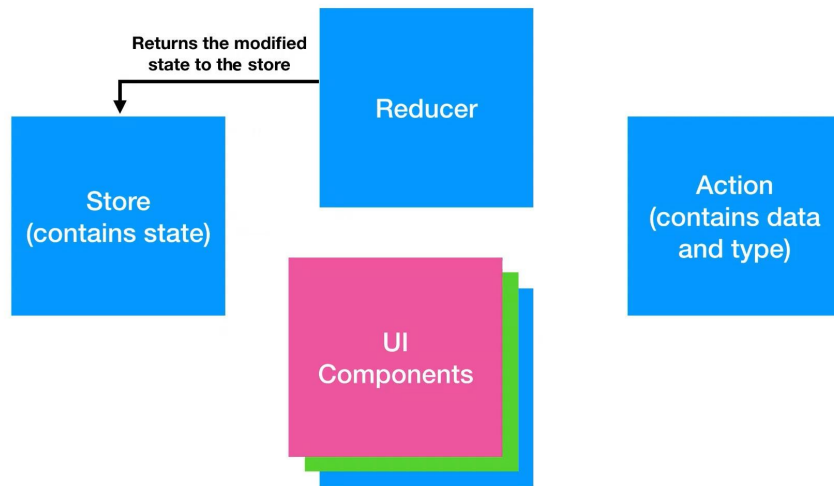
# Data flow in Redux

- The action is transferred to Reducer, which receives two parameters, actions and initial state from store.



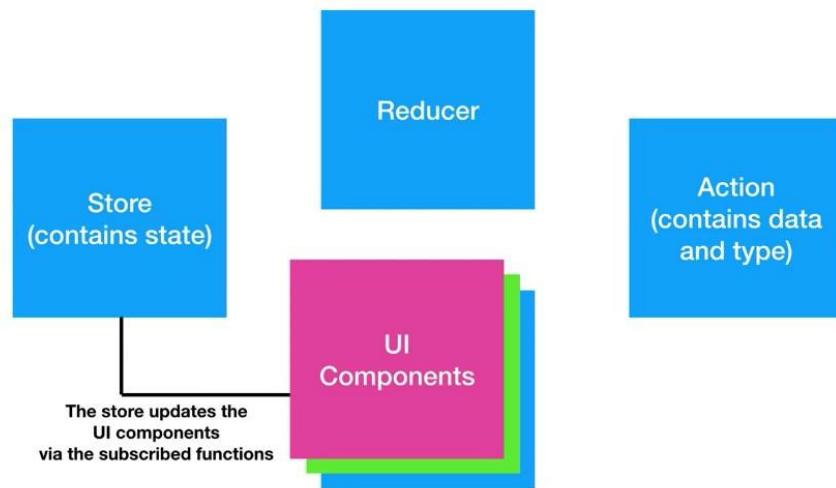
# Data flow in Redux

- The Reducer compares the initial state with the action dispatched, updates the state based on action type and returns the modified state to store.



# Data flow in Redux

- The store then updates the states and renders the updated UI components.



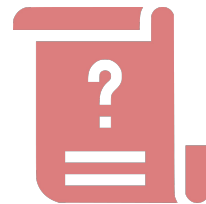
# Creating our Form

We will be creating a Redux form using the concepts we have learnt so far.

<https://codesandbox.io/s/wandering-sun-ev244>

# Assignment

Expand the form to add more fields like date of birth, city, state, country and add meaningful validations.



**Thank You**