Redux

Presented by: Hau Nguyen

NORDIC CODER

Agenda

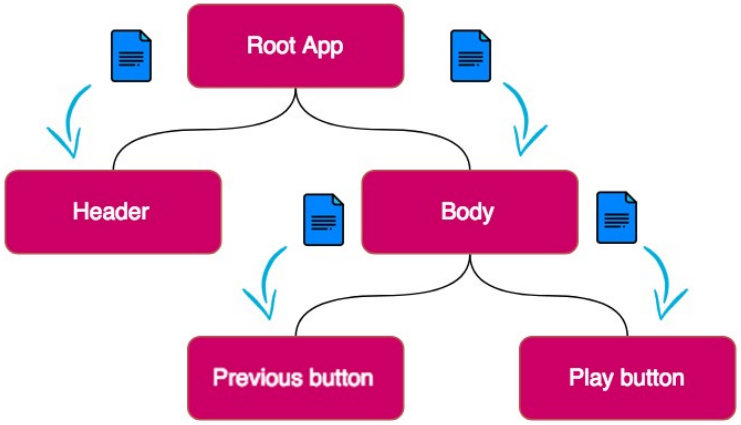
- React Context API
- Redux keywords
- Three Principles
- When to use Redux
- Demo
- Q & A

NORDIC
< CODER >

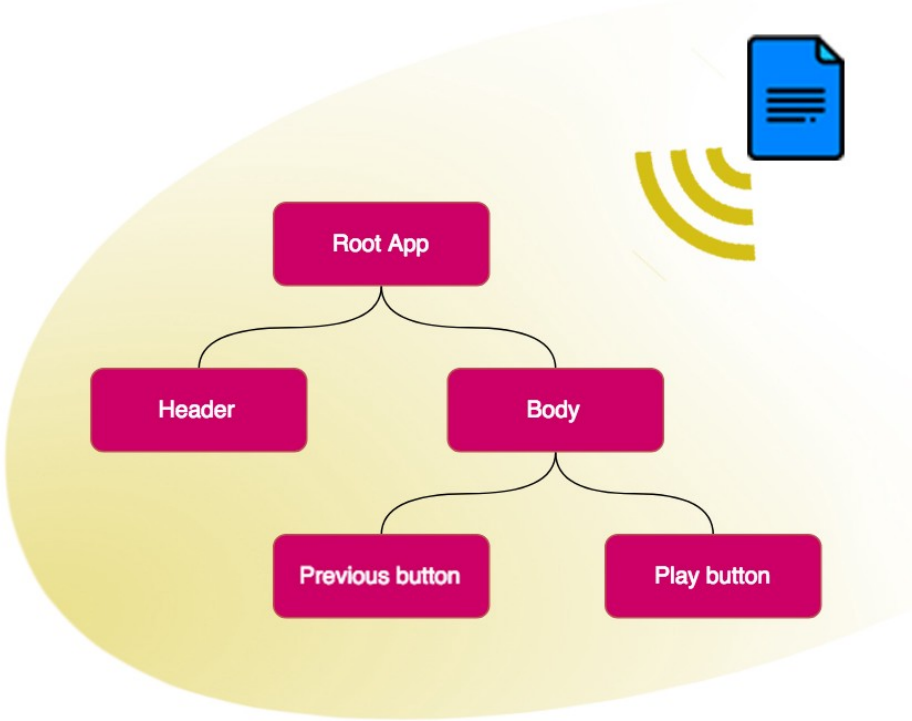
React Context API

NORDIC
< CODER >

Why React Context?



Passing prop to each level



The data in Context available to every component

Provider

Consumer

```
const ThemeContext = React.createContext();
class App extends PureComponent {
  constructor(props) {
    super(props);
    this.state = {
      theme: {
        primaryColor: 'deeppink',
      }
    };
  }

  render() {
    return (
      <ThemeContext.Provider value={this.state.value}>
        <Paragraph text="Hello World" />
      </ThemeContext.Provider>
    );
  }
}

// ----------

class Paragraph extends PureComponent {
  render() {
    return (
      <ThemeContext.Consumer value={this.state.value}>
        {theme => (
          <p style={{ color: theme.primaryColor }}>{this.props.text}</p>
        )}
      </ThemeContext.Consumer>
    );
  }
}
```

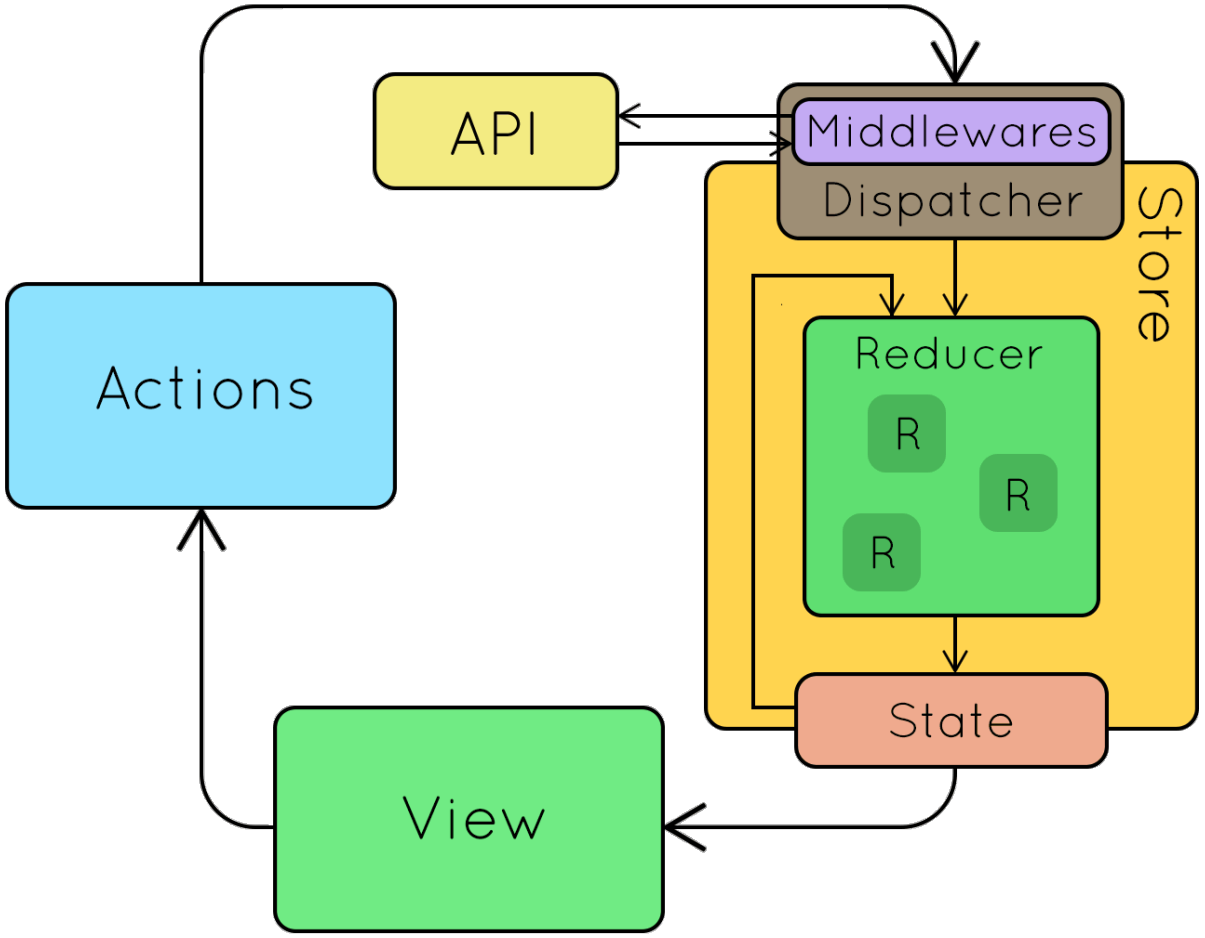
NORDIC CODER

Key words

- **Store**
- **State**: current value of store
- **Action**: the only way to change value of state
- **Reducer**: define how state is changed by action
- **dispatch**

NORDIC
< CODER >

Redux flow



NORDIC CODER

Three principles

- Single source of truth
- State is read-only
- Changes are made with pure functions

3 fundamental principles for Redux

One Immutable Store for All the State

Actions trigger changes

All state changes are made by pure functions i.e. Reducers

When to use Redux

- Global state management
- State shared between components
- Time series tracking

Redux simple example

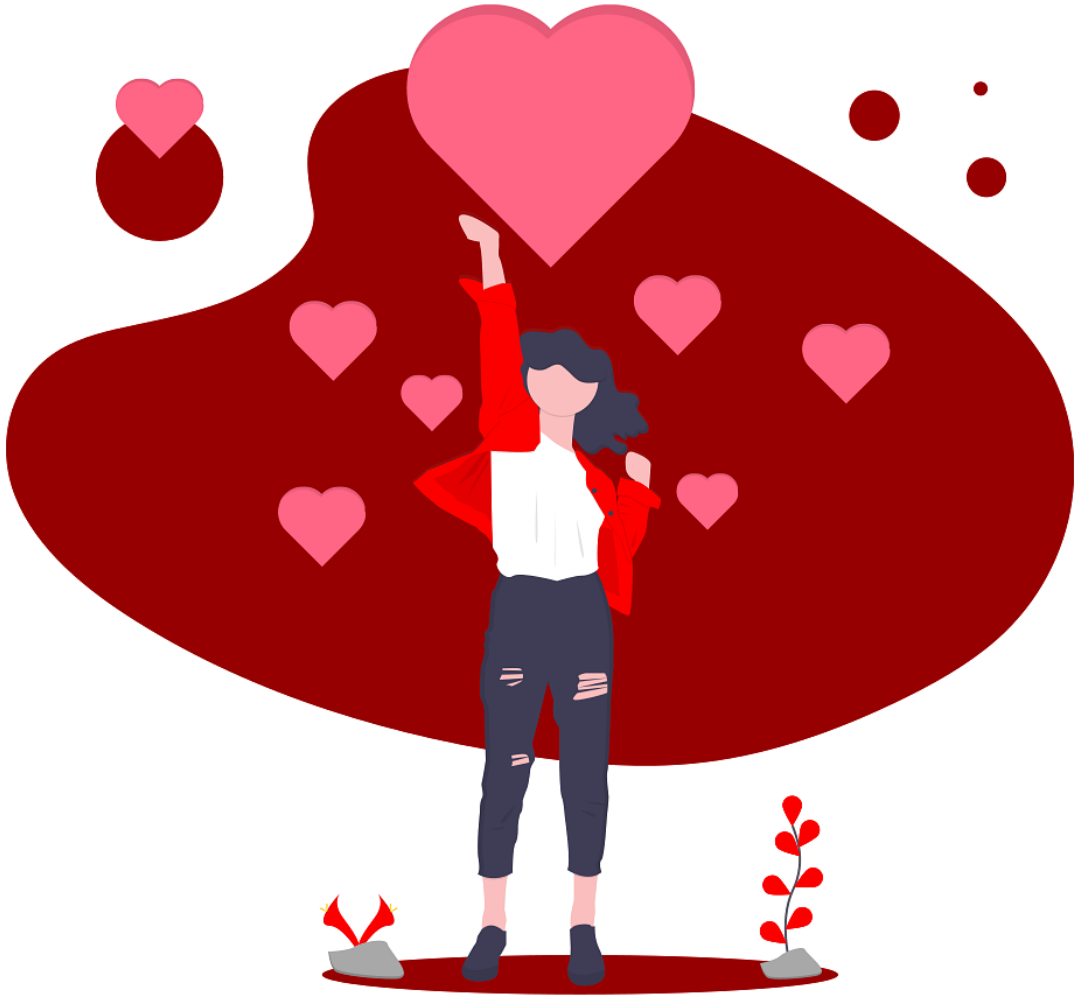
```
import { createStore } from 'redux'

function counter(state = 0, action) {
  switch (action.type) {
    case 'INCREMENT':
      return state + 1
    case 'DECREMENT':
      return state - 1
    default:
      return state
  }
}

let store = createStore(counter)
store.subscribe(() => console.log(store.getState()))
store.dispatch({ type: 'INCREMENT' })
store.dispatch({ type: 'DECREMENT' })
```

NORDIC CODER


LIVE DEMO

NORDIC
< CODER >

Thank You
And see you ;)



NORDIC
< CODER >