

Episode 11 - Data is the New Oil

Layers of Web App :-

1. UI layers → Everything you see on the screen (the view)
2. Data layers → Maintained by states & props

JSX → JS → Object → Reconciliation
via visible in
Babel Virtual DOM

The updation b/w previous Virtual Dom & new Virtual DOM is diff

Difference b/w State & props ?

State is a local variable in a component props is the value passed from one component to other component.

⇒ Props drilling

App Layout (state = user)

- <Body user = {user} />
- <RestaurantContainer user = {user} />
 - <RestaurantCard user = {user} />
 - <h4> {user} </h4>

Sending data from higher hierarchy to down the chain to children
But the data flows one way

How to send data from child to parent components ?

1. Custom Hooks
2. Redux

Is prop drilling a good way ?

1. No, it makes the code cluttered as we pass the data down the line but sometimes in some components we don't even need that information but needs to be passed to the child below
2. It has unnecessary variables which will be rendered again & again, it makes it difficult to debug in React dev tools



→ Lifting the State up

We can't change/control a sibling component so we have to lift the state to the parent so that all the children can be controlled through parent component

React Devtools

Profiler < Flamegraph

You can rank the components based on their loading time

React context

If you want a data all across your app, then props drilling is not a good way instead you should use Context API or Redux, MobX, NGRX, Flux

Why you should not make a global variable to pass it in any component?

Because React will not track its state. No reconciliation

How to create context?

```
import { createContext } from "react";
```

Context is a function at the end of the day.

Difference between useState/props and context

useState / props are tied to a component but context is not. Context can be stored in a central store which is not specific to any component. Context is like useState for whole app

Context in Class Component

```
<UserContext.Consumer>
  {({user}) =>
    <h4> {user.name} - {user.email} </h4>
  }
</UserContext.Consumer>
```

How to override default value of context

```
const [user, setUser] = useState()
```

```
<UserContext.Provider
  value={user}
>
  <Header/>
  <Outlet/>
  <Footer/>
</UserContext.Provider>
```

When creating a context you can give names to debug it in React DevTool?

Even when the code is not there (Lazy Loading). The Virtual Dom will be updated & the data will be displayed on the DOM.