



Section 10: React's Context API & useReducer - Advanced State Management

23 September 2024 09:36



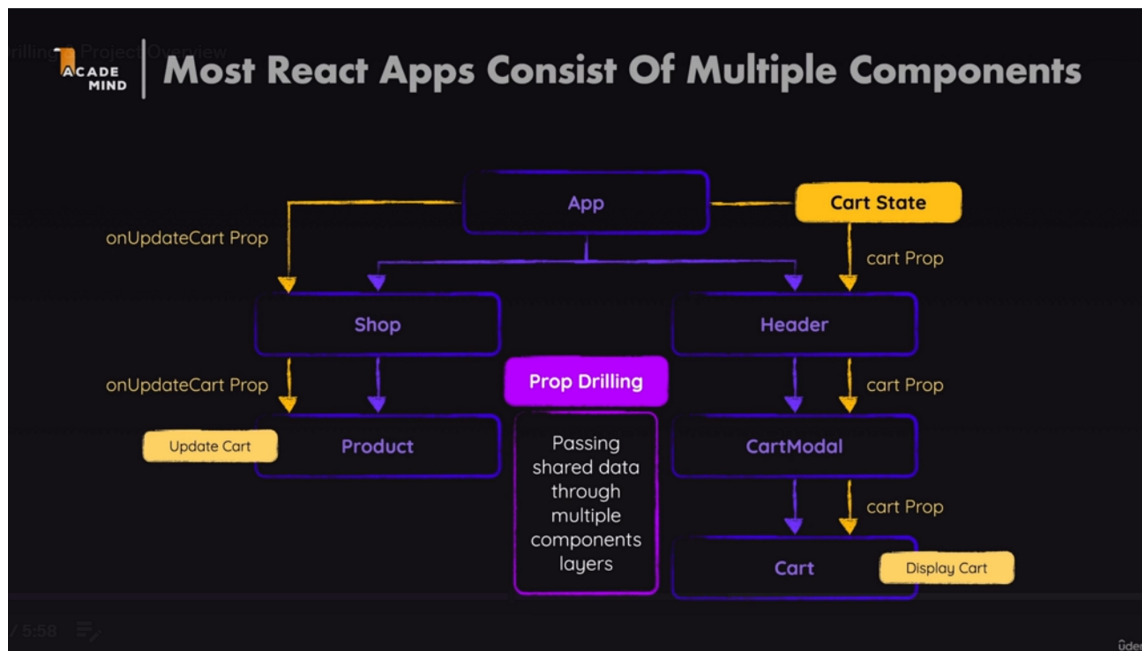


Advanced State Management

Beyond Basic Apps & "Lifting Up State"

- ▶ The Problem Of Shared State: Prop Drilling
- ▶ Embracing **Component Composition**
- ▶ Sharing State with **Context**
- ▶ Managing Complex State with **Reducers**

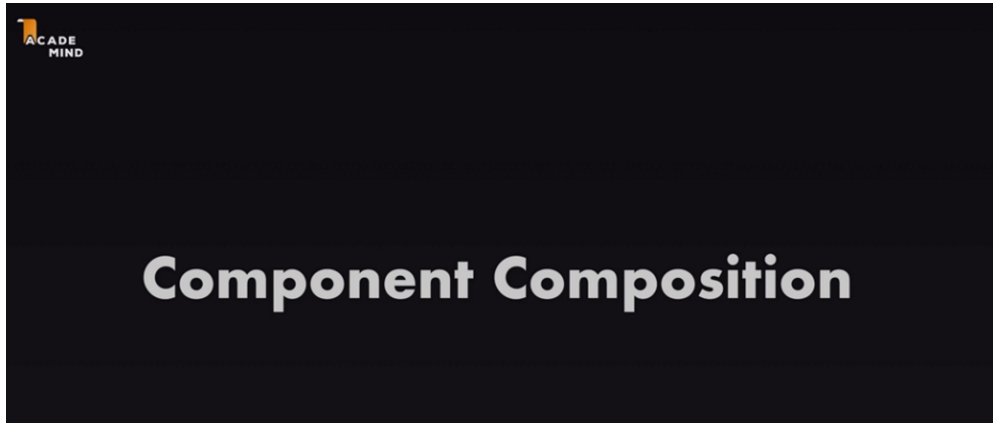
Udemy



Prop Drilling: Component Composition as a Solution

Now one possible solution for this drilling problem is to embrace component composition. And I will say right away that this can be part of the overall solution, that this can solve prop drilling

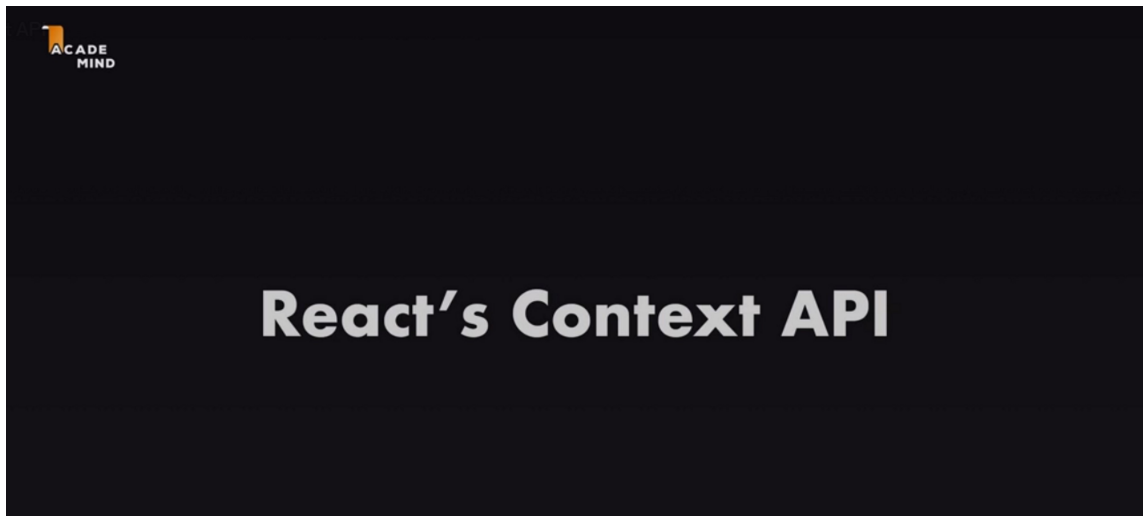
in certain situations, but that in most cases, you'll also need that second solution we'll take a look at later.



Typically you don't want this solution for all your component layers. Because it would mean that in the end, all your components just end up in the app component.

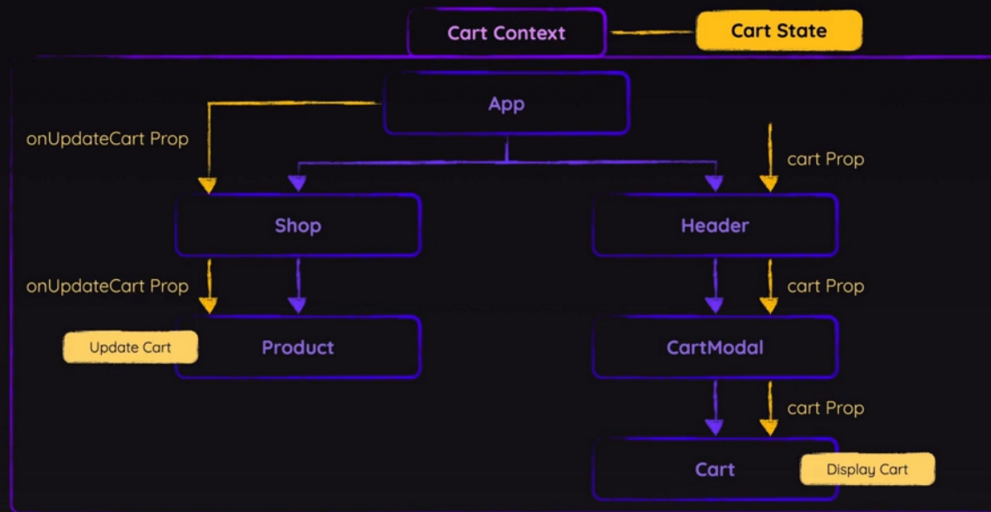
Introducing the Context API

So with component composition being one possible solution, or as mentioned a part of an overall solution, let's take a look at the other possible solution. A solution that can indeed get rid of this entire prop drilling problem altogether. And that would be React's Context API which is another feature built into React. And it is a feature that's there to make sharing data across Components and across component layers a breeze.



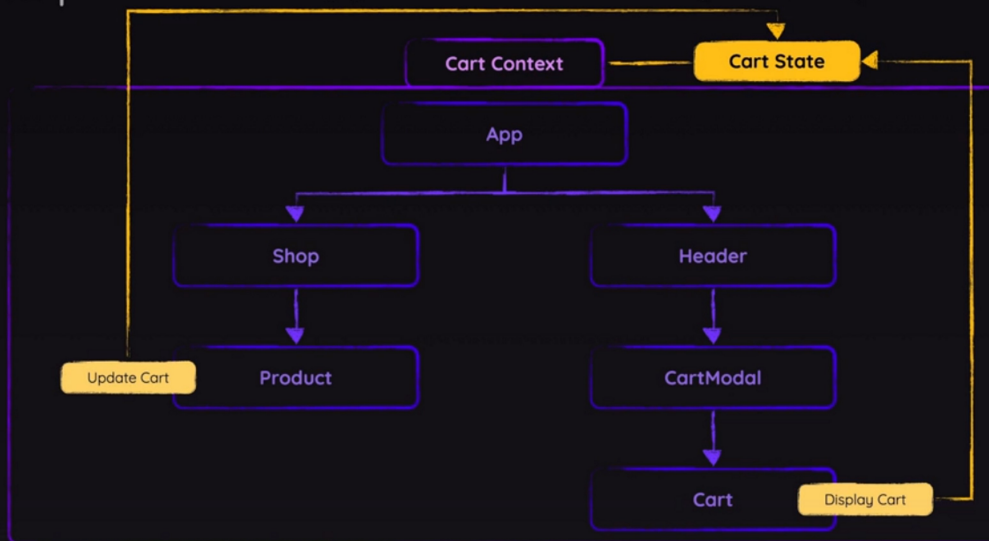
And the idea behind this context feature offered by React is that you create a context value and that you then provide that value that you wrap this context, you could say around multiple Components, possibly around all Components of your app.

Most React Apps Consist Of Multiple Components



Udemy

Most React Apps Consist Of Multiple Components



Udemy

Now, the great thing about this context value that you provide to multiple Components is that it can easily be connected to State. So, you can

connect your React state to that context value which is provided to the entire application. And therefore, you can get rid of all those props. You don't need to pass the state or the state updating functions through multiple component layers anymore. Instead, since that context value, which is linked to State is provided to all Components in your application, the Components that do need access, that do need to change or read the State can directly reach out to that context, and therefore, also to that State. And that's therefore a very powerful solution which we'll now explore in practice in greater detail.

Introducing the useReducer Hook

