Simpler State Management with the Context API

In this lesson, we'll look at particular situations where we can use the Context API.

WE'LL COVER THE FOLLOWING

- An Efficient Solution

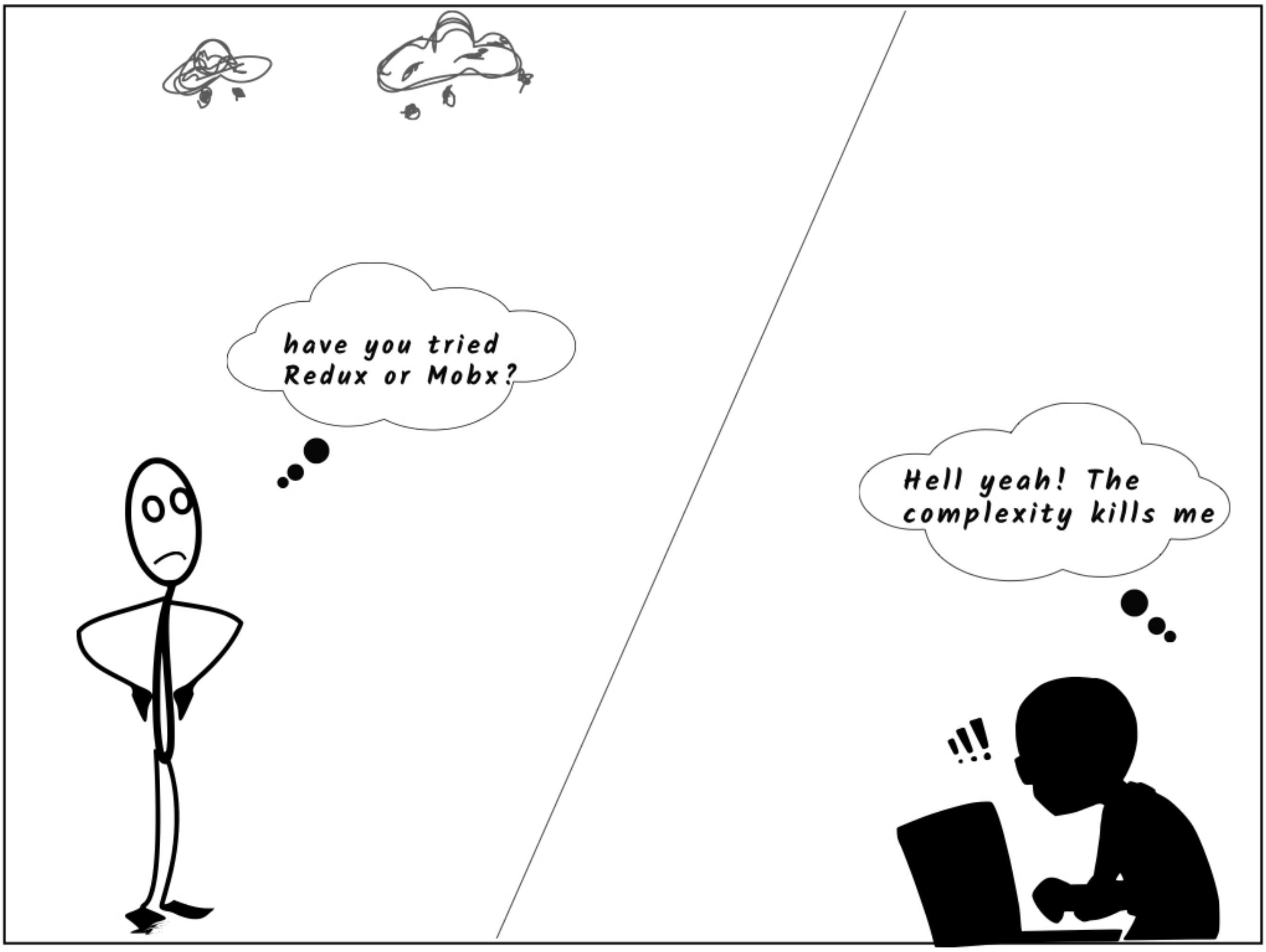
John’s an amazing developer, and he loves what he does. However, he’s been facing a recurring problem when writing React applications.

Props drilling

Props drilling, the term used to describe passing down props needlessly through a deeply nested component tree, has plagued John for a while now!

Luckily, John has a friend who always has the answers. Hopefully, she can suggest a way out.

John reaches out to Mia, and she steps in to offer some advice.


have you tried Redux or Mobx?
Hell yeah! The complexity kills me
!!!

Mia says, 'try Redux or MobX'

Mia is a fabulous engineer as well, and she suggests using a state management library such as `Redux` or `MobX`.

These are great choices, however, for most of John’s use cases, he finds them a little too bloated.

John asked, “Can’t I have something simpler and native to React itself?”

An Efficient Solution

Mia goes on a desperate search to help a friend in need, and she finds the Context API.

Mia recommends using React's Context API to solve the problem. John is now happy and excited to see what the Context API could offer, and he goes about his work productively.

This marks the beginning of John's experience with the Context API.

In the next lesson, we'll finally be introduced to Context.