React Native Hooks

What is Hooks?

Hooks let you use state and other React features without writing a class. Hooks are functions that let you “hook into” React state and lifecycle features from function components.

When would I use a Hook?

If you write a function component and realize you need to add some state to it, previously you had to convert it to a class. Now you can use a Hook inside the existing function component.

Why Hooks?

1. Complex components become hard to understand!
   * it’s not possible to break these components into smaller ones because the stateful logic is all over the place. It’s also difficult to test them. However, that often introduces too much abstraction, requires you to jump between different files, and makes reusing components more difficult.
   * Hooks let you split one component into smaller functions based on what pieces are related (such as setting up a subscription or fetching data), rather than forcing a split based on lifecycle methods.

React provides a few built-in Hooks like useState. You can also create your own Hooks to reuse stateful behavior between different components.

Let us start with **useState.**

# useState

What does calling useState do?

It declares a “state variable”. In our example it will be firstName and lastName.

What do we pass to useState as an argument?

It returns pair of values: current state (or value) and a function that can be used to update its value.

Let’s make this simple form, to enter the first name & last name.

So, previously to do this we should do these steps:

1. Definition the first name and the last name as a state and initialize them

**constructor**(props) {  
 super(props);  
 this.state =   
 {  
 name: 'Sohad',  
 surename: 'Dader'  
 };  
}

1. And then we add them as a value for the TextInput and its text change event

<**TextInput**  
 placeholder='First Name'  
 style={input}  
 value={firstName}  
 onChangeText={(text) => this.setState({firstName: text})}  
 />

This is the traditional way.

Now let’s see how can we do same thing with useState hook.

1. First import useState

**import** React, {useState} **from** 'react';

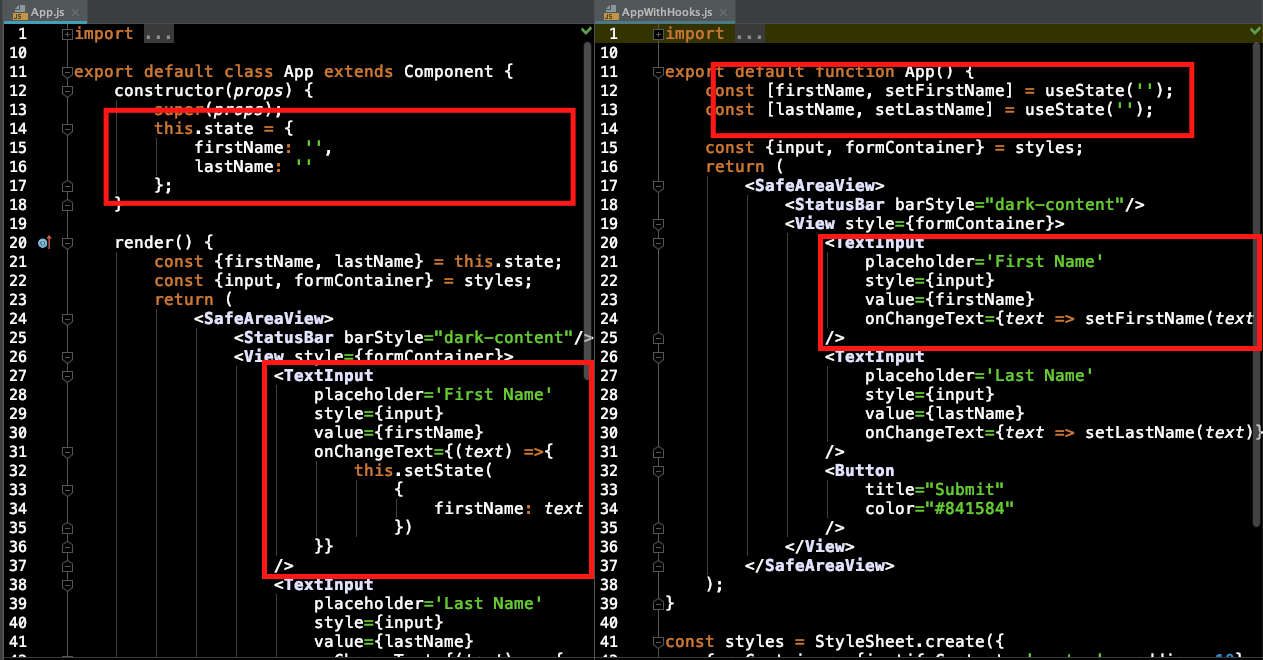
1. Now we can create two fields using useState like this

**const** [firstName, setFirstName] = useState('');  
**const** [lastName, setLastName] = useState('');

1. Now we can use it something like this

<**TextInput**  
 placeholder='First Name'  
 style={input}  
 value={firstName}  
 onChangeText={text => setFirstName(text)}  
 />

let see the difference between them on the code



https://miro.medium.com/max/60/1*OvX7b7Ti518HYiAHzcNkEA.png?q=20