

Introdução ao React.JS

Othon Oliveira

SENAC - PROA



Sumário

- 1 Respondendo a eventos no React
- 2 Ganchos ou Hooks?
- 3 Adicionando “estado”
- 4 Upgrade ao componente ContaBancaria



Minha Imagem



Figura: Componentes React



O que são eventos?

Um simples click no botão

Handle event

Você pode responder a eventos declarando funções de manipulador de eventos (handle event) dentro de seus componentes:



ClickComponente.js

```
1 function MeuBotao() {  
2   function manipulaClick() {  
3     alert('Voce clicou em mim!');  
4   }  
5  
6   return (  
7     <button onClick={manipulaClick}>  
8       Click em mim  
9     </button>  
10  );  
11 }
```

Bora fazer?

Observe como `onClick={manipulaClick}` não tem parênteses no final! Não chame a função do manipulador de eventos: você só precisa transmiti-la. O React chamará seu manipulador de eventos quando o usuário clicar no botão.

O que é um Hook?

Traduzindo ao pé da letra o termo

Gancho

Hook: Um dispositivo curvado, metálico, na forma de um gancho; Hook também pode ser um termo usado nas lutas de boxe, gancho de esquerda (ou direita): a left/right hook



O que é um Hook?

Traduzindo ao pé da letra o termo

Gancho

Hook: Um dispositivo curvado, metálico, na forma de um gancho; Hook também pode ser um termo usado nas lutas de boxe, gancho de esquerda (ou direita): a left/right hook

Hook em React

Os hooks (ganchos) em React são funções que permitem "conectar-se" ao estado do React e aos recursos do ciclo de vida a partir de componentes.



O que é um Hook?

Traduzindo ao pé da letra o termo

Gancho

Hook: Um dispositivo curvado, metálico, na forma de um gancho; Hook também pode ser um termo usado nas lutas de boxe, gancho de esquerda (ou direita): a left/right hook

Hook em React

Os hooks (ganchos) em React são funções que permitem "conectar-se" ao estado do React e aos recursos do ciclo de vida a partir de componentes.



Figura: A volta do Capitão Gancho

Tipos de hooks

Existem dois tipos principais

useState

Este hook (gancho) permite adicionar estado aos seus componentes funcionais. Ele pega um valor de estado inicial e retorna um array com o valor do estado atual e uma função para atualizar o estado.



Tipos de hooks

Existem dois tipos principais

useState

Este hook (gancho) permite adicionar estado aos seus componentes funcionais. Ele pega um valor de estado inicial e retorna um array com o valor do estado atual e uma função para atualizar o estado.

useEffect

Este gancho permite que você execute efeitos “colaterais”, como busca de dados, quando seu componente é montado ou atualizado.



Criando o componente Hook

Mudança de estado - Um simples click no botão

Handle event

primeiro vamos importar “useState” do React

```
1 import { useState } from 'react';
```

Agora você pode declarar uma variável de estado dentro do seu componente:

```
1 function MeuBotao() {  
2   const [contar, setContar] = useState(0);  
3   // ...  
4 }
```



Quem são os manipuladores?

[“algumaCoisa”, “setAlgumaCoisa”]

Na primeira vez que o botão for exibido, a contagem será 0 porque você passou 0 para `useState()`. Quando você quiser mudar de estado, chame `setContar()` e passe o novo valor para ele. Clicar neste botão aumentará o contador:

```
1 function MeuBotao() {  
2   const [contar, setContar] = useState(0);  
3  
4   function manipulaClick() {  
5     // ...  
6   }  
7 }
```



Mundando o estado de um componente

Bora fazer?

Na primeira vez que o botão for exibido, a contagem será 0 porque você passou 0 para `useState()`. Quando você quiser mudar de estado, chame `setContar()` e passe o novo valor para ele. Clicar neste botão aumentará o contador:

```
1 function MeuBotao() {  
2   const [contar, setContar] = useState(0);  
3  
4   function manipulaClick() {  
5     setContar(contar + 1);  
6   }  
7  
8   return (  
9     <button onClick={manipulaClick}>  
10      Clicou {contar} vezes  
11    </button>  
12  );  
13 }
```



Upgrade

Bora fazer?

```
1 import React, { useState } from 'react';
2 import ContaBancaria from './ContaBancaria';
3
4 function App() {
5   const [saldo, setSaldo] = useState(1000); // Valor inicial do saldo
6
7   const atualizarSaldo = (novoSaldo) => {
8     setSaldo(novoSaldo);
9   };
10  return (
11    <div>
12      <h1>Minha Aplicacao Bancaria</h1>
13      <ContaBancaria saldoInicial={saldo} atualizarSaldo={atualizarSaldo} />
14    </div>
15  );
16 }
17 export default App;
```

