Aula 2 - Hooks

Docupedia Export

Author:Ferro Alisson (CtP/ETS)
Date:24-Aug-2023 13:44

Table of Contents

1 useState

Até o presente momento estávamos criando páginas estáticas, mas embora podemos fazer isso, o React vem para facilitar para nós criarmos páginas dinâmicas, para isso temos os chamados Hooks do React
Segundo a documentação, *Hooks* são uma nova adição no React 16.8. Eles permitem que você use o estado e outros recursos do React sem escrever uma classe. Isso significa que antes dos Hooks, precisávamos criar nossos componentes dentro de classes, e agora nosso componente pode ser uma função, o que simplifica a escrita.
Nosso primeiro Hooks que iramos ver, será o *useState*. O useState permite alterar o estado de uma variável. Para isso basta importar o useState e colocar a variável entre colchetes e o valor inicial da variável vai como parâmetro do useState, como a seguir

```
import { useState } from "react"

export default function Counter(){
    var [num, setNum] = useState(0);

    return (
        <>
            {num}
        </>
    )
}
```

o *num* é nossa variável que queremos alterar com o *useState*, *setNum* é a função que altera o *num*, sendo assim, para alterar vamos adicionar um botão de soma para fazer um contador, ficando com o codigo a seguir

```
import { useState } from "react"

export default function Counter(){
    var [num, setNum] = useState(0);

    return (
        <>
            <button onClick={handleZero}>Clique aqui</button>
            {num}
            <button onClick={handleFive}>Clique aqui</button>
```

```
        </>
    )
}
```

o *onClick* é um um evento do DOM, significa que será manipulado o DOM toda vez que o elemento *button* for clicado.
Precisamos criar nossa função handleFive e nossa função handleZero

```
function handleFive(){
    setNum(5);
}

function handleZero(){
    setNum(0);
}
```

1.1 Desafio 1: crie um novo componente e faça um contador com dois botões, um que incremente e outro que decremente o num e aplique um estilo ao contador.

2 onChange

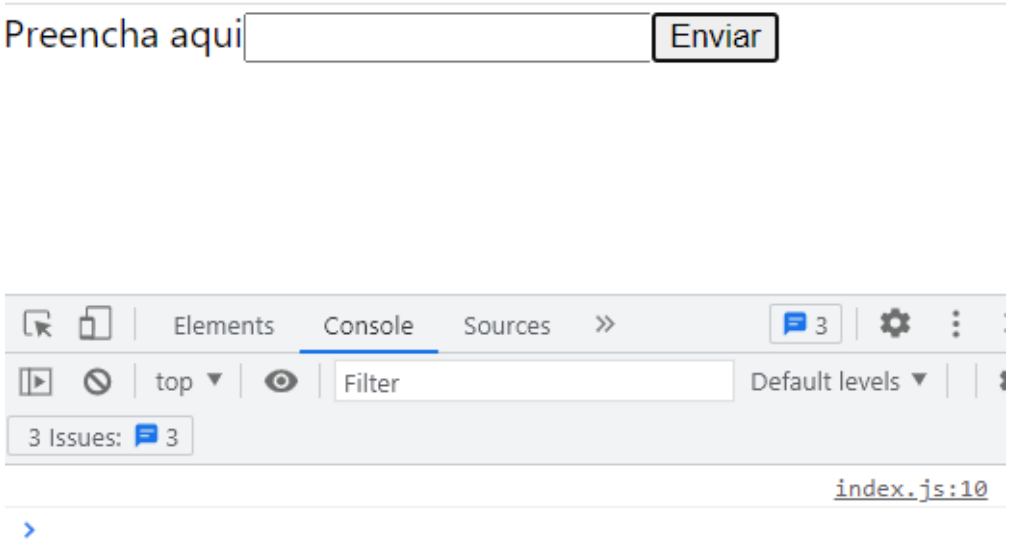
Já vimos um evento do DOM que foi o onClick, agora vamos ver mais um evento que é o onChange.
Esse evento é chamado toda vez que um valor no input for alterado, assim, se tivermos um formulario podemos usar o useState e o set será chamado toda vez que o valor for alterado.
Para isso vamos criar um novo componente *src/components/Form/index.js* e vamos adicionar nosso componente

```
import { useState } from "react"

export default function Counter(){
    var [text, setText] = useState('');

    return (
        <>
            <label>Preencha aqui</label>
            <input />
            <button onClick={() => console.log(text)}>Enviar</button>
        </>
    )
}
```

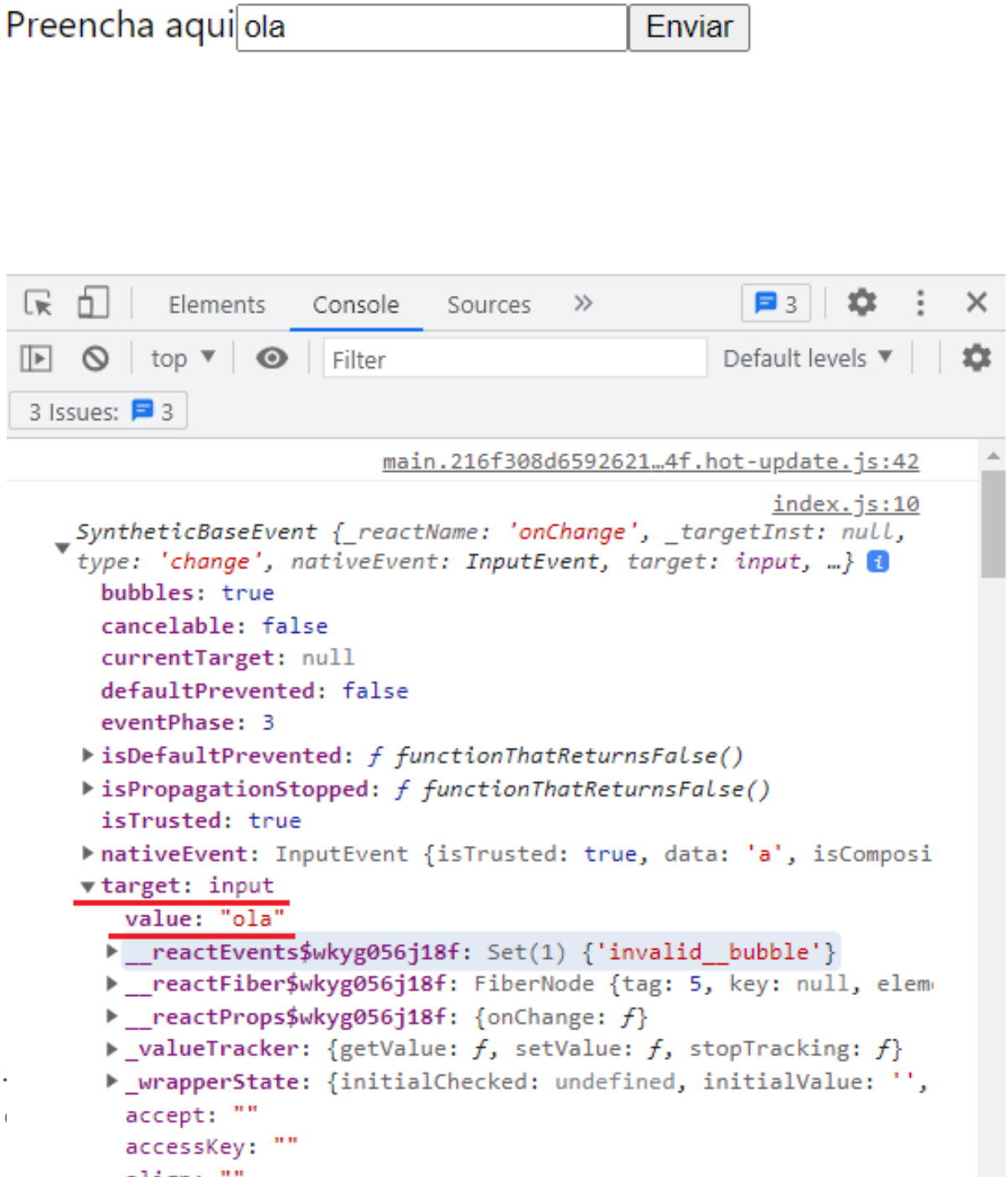
Se clicarmos no botão da forma que está, no console irá imprimir uma string vazia, que foi o que nós usamos no useState como valor default para a variável *text*



Agora vamos ver como o onChange funciona, para isso vamos adicionar o seguinte código no input.

```
onChange={(e) => console.log(e)}
```

Com isso, a cada letra que digitarmos, será chamada a função anônima acima exibir no console. Se nós expandirmos o conteúdo vamos observar que se expandir, terá *target* e dentro de *target* tem *value* onde corresponde ao valor que nós digitamos no input



Mas isso ainda não altera a variável, precisamos usar o onChange para modificar nossa variavel *text,* para isso vamos colocar o código no input

```
onChange={(e) => setText(e.target.value)}
```

Agora sim a variável text esta sendo alterada, para ver isso basta clicar em enviar, onde irá imprimir o conteúdo da variável *text*

2.1 Desafio 2: crie uma calculadora com soma, subtração, multiplicação, divisão.

3 useEffect

O useEffect é nosso segundo Hook que iremos ver, é um Hook que podemos utilizar de várias formas, ele é possui dois parâmetros, o primeiro é o método que é chamado quando algo ser alterado no array de dependências, e o segundo parâmetro é o array de dependências, nesse array, nós passamos as variáveis que serão monitoradas, quando altera o valor, o método que está como parâmetro é chamado.
Resumindo, toda vez que nossa variável é alterada executamos o método.
vamos utilizar o formulário da calculadora, para toda vez que nosso num1 ou num2 forem alterados, já realize o cálculo sem precisar clicar no botão

```
import { useEffect, useState } from 'react';

function App() {
  var [num, setNum] = useState(0);
  var [num2, setNum2] = useState(0);
  var [soma, setSoma] = useState(0);

  useEffect(() => {
    setSoma(num+num2)
  }, [num, num2])

  return (
    <>
      <form>
        <input
          type='number'
          value={num}
          onChange={(e) => setNum(Number(e.target.value))}
        />
        <input
          type='number'
          value={num2}
          onChange={(e) => setNum2(Number(e.target.value))}
        />
        Soma: {soma}
      </form>
    </>
  );
}
```

```
export default App;
```

4 useRef

É um hook do React, usado para acessar e interagir com elementos do DOM ou para persistir valores entre re-renderizações de um componente, sem acionar uma re-renderização adicional.
Pode ser utilizado para acessar componentes quando determinadas coisas ocorrerem.
Exemplo:

```
import { useState, useRef } from 'react';

function App() {
    const [name, setName] = useState('');
    const refInp = useRef(null);

    function handleName(){
        if(name.length<3)
            refInput.current.focus();
    }

  return (
    <>
      <form>
        <input
          ref={refInput}
          onBlur={handleName}
          type='text'
          value={name}
          onChange={(e) => setNum(Number(e.target.value))}
        />
      </form>
    </>
  );
}

export default App;
```

Ou seja, o onBlur é um evento do DOM que é chamado toda vez que o input perde o foco, sendo assim chamamos a função handle name, e caso o nome for menor que 3 caracteres, o foco será dado ao input novamente

4.1 Desafio 3: Implemente um contador que aumente a cada 5 segundo passado e altere e titulo da página e que exiba uma notificação na tela sempre que o título da página for alterado, não pode ser alert().