

Aula 5 - useContext

Docupedia Export

Author:Ferro Alisson (CtP/ETS)

Date:29-Aug-2023 15:51

Table of Contents

1 Desafio: Crie uma tela seguindo o padrão do figma a seguir, onde são duas telas, desenvolva em SPA e passando as informações via context para o extrato, implemente também o dark mode no site.

7

Até o momento vimos alguns *Hooks* do React, agora iremos ver mais um, o `useContext`.

Do latim *contextus*, o contexto é o ambiente físico ou situacional (conjunto de circunstâncias) a partir do qual se considera um fato.

Contextos são utilizados para compartilhar informações entre os componentes, sendo assim, um componente pode alterar informações para serem refletidas em outro componente sem grande complexidade,

Até o momento para passar informações nós passávamos via *props*, agora com o contexto será mais simples.

Para utilizar, vamos voltar para o nosso projeto do contador, mas vamos realizar algumas alterações. Até o momento nosso componente era assim:

```
import { useState } from "react"

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

  function handleDown(){
    setNum(--num)
  }

  function handleUp(){
    setNum(++num)
  }

  return (
    <>
      <button onClick={handleDown}>Clique aqui</button>
      {num}
      <button onClick={handleUp}>Clique aqui</button>
    </>
  )
}
```

Vamos criar um arquivo em `src/context/counter.js` e lá vamos criar nosso contexto e colocar a lógica de incrementar ou decrementar. Para criar o contexto, vamos adicionar o código

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

export const CounterContext = React.createContext();
CounterContext.displayName = 'Counter';
```

Onde criamos o contexto com `React.createContext()` e atribuímos o nome do contexto em `CounterContext.displayName`; Após isso, é necessário criar nosso `Provider`, no mesmo arquivo, vamos colocar o código

```
export const CounterProvider = ({ children }) => {  
  const [num, setNum] = useState(0);  
  
  function handleDown(){  
    setNum(num - 1)  
  }  
  
  function handleUp(){  
    setNum(num + 1)  
  }  
  
  return(  
    <CounterContext.Provider  
      value={{  
        num,  
        setNum,  
        handleDown,  
        handleUp  
      }}  
    >  
      {children}  
    </CounterContext.Provider>  
  )  
}
```

Assim criamos nosso *Provider*, onde pegamos a lógica de incrementar e decrementar e colocamos no *provider*. *Falta somente uma ultima configuração para que possa ser usado o contexto. Em src/App.js adicionaremos o Provider envolvendo todas as rotas.*

```
import './App.css';  
import { CounterProvider } from './context/counter';  
import FormPage from './pages/form';  
import CounterPage from './pages/counter';
```

```
import NavBar from './components/NavBar';
import { Routes, Route, Navigate } from "react-router-dom";
import 'bootstrap/dist/css/bootstrap.min.css';

function App() {
  return (
    <>
      <CounterProvider>
        <NavBar />
        <Routes>
          <Route path='/counter' element={<CounterPage />} />
        </Routes>
      </CounterProvider>
    </>
  );
}

export default App;
```

Agora sim terminamos de configurar o contexto, para usar o contexto basta ir na página que queremos utilizar e importar, como a seguir

```
import { useState, useContext } from "react"
import { CounterContext } from '../context/counter';

export default function Counter(){
  const { handleDown, handleUp } = useContext(CounterContext)

  return (
    <>
      <button onClick={handleDown}>Clique aqui</button>
      <button onClick={handleUp}>Clique aqui</button>
    </>
  )
}
```

Nesse componente só precisamos *handleDown* e *handleUp*, portanto serão somente estes que importaremos. No componente onde será exibido o *num* precisamos importar somente ele, ficando com o código

```
import { useContext } from "react"
import { CounterContext } from "../../context/counter"

export default function Numbers(){
  const { num } = useContext(CounterContext)
  return (
    <>{num}</>
  )
}
```

E agora no *CounterPage* vamos importar esses components

```
import Counter from '../components/Counter';
import Numbers from '../components/Numbers';

export default function CounterPage(){
  return(
    <>
      <Counter />
      <Number />
    </>
  )
}
```

1 Desafio: Crie uma tela seguindo o padrão do figma a seguir, onde são duas telas, desenvolva em SPA e passando as informações via context para o extrato, implemente também o dark mode no site.

The image displays two mobile application screens side-by-side. The left screen features a top navigation bar with three items: 'Home', 'Extrato', and 'Adicionar Pagamento'. The right screen features a top navigation bar with a hamburger menu icon. Both screens contain three text input fields, each with the placeholder text 'Digite o texto', and a blue button labeled 'Salvar' at the bottom.

HomeExtratoAdicionar Pagamento

Carteira	Data	Valor	Saldo
Carteira	1/1/2023	0,00	0,00
Carteira	1/1/2023	0,00	0,00
Carteira	1/1/2023	0,00	0,00
Carteira	1/1/2023	0,00	0,00
Carteira	1/1/2023	0,00	0,00
Carteira	1/1/2023	0,00	0,00
Carteira	1/1/2023	0,00	0,00
Carteira	1/1/2023	0,00	0,00
Carteira	1/1/2023	0,00	0,00
Carteira	1/1/2023	0,00	0,00
Carteira	1/1/2023	0,00	0,00
Carteira	1/1/2023	0,00	0,00

Carteira	Data	Valor	Saldo
Carteira	1/1/2023	0,00	0,00
Carteira	1/1/2023	0,00	0,00
Carteira	1/1/2023	0,00	0,00
Carteira	1/1/2023	0,00	0,00
Carteira	1/1/2023	0,00	0,00
Carteira	1/1/2023	0,00	0,00
Carteira	1/1/2023	0,00	0,00
Carteira	1/1/2023	0,00	0,00
Carteira	1/1/2023	0,00	0,00
Carteira	1/1/2023	0,00	0,00
Carteira	1/1/2023	0,00	0,00
Carteira	1/1/2023	0,00	0,00

Adicionar Pagamento