

# Criando o contexto - Counter

useContext

src/hooks/useCounter

```
1 import { createContext, useContext, useState, } from "react";
2
3 const CounterContext = createContext({});
4
5 export function CounterProvider({ children }) {
6   const [count, setCount] = useState(0);
7
8   function increment() {
9     setCount((count) => count + 1);
10  }
11
12  function decrement() {
13    setCount((count) => count - 1);
14  }
15
16  function incrementByAmount(amount) {
17    setCount((count) => count + amount);
18  }
19
20  return (
21    <CounterContext.Provider
22      value={{
23        count,
24        increment,
25        decrement,
26        incrementByAmount,
27      }}
28    >
29      { children }
30    </CounterContext.Provider>
31  )
32 }
33
34 const useCounter = () => useContext(CounterContext);
35
36 export default useCounter;
37
```

RTK

src/slices/counter

```
1 import { createSlice } from "@reduxjs/toolkit";
2
3 const initialState = {
4   value: 0,
5 };
6
7 const reducers = {
8   increment: (state) => {
9     state.value += 1;
10  },
11
12   decrement: (state) => {
13     state.value -= 1;
14  },
15
16   incrementByAmount: (state, action) => {
17     state.value += action.payload;
18  },
19 };
20
21 export const counterSlice = createSlice({
22   name: 'counter',
23   initialState,
24   reducers,
25 });
26
27 export const { increment, decrement, incrementByAmount } = counterSlice.actions;
28
29 export default counterSlice.reducer;
30
```

# Criando o contexto - User

useContext

src/hooks/useUser

RTK

src/slices/user

```
1 import { createContext, useContext, useState } from 'react';
2
3 const UserContext = createContext({});
4
5 export function UserProvider({ children }) {
6   const [name, setName] = useState('');
7   const [githubUsername, setGithubUsername] = useState(null);
8   const [highestCount, setHighestCount] = useState(0);
9
10  // wrappers para poder adicionar funcionalidade caso necessario sem modificar a interface;
11  const updateName = setName;
12  const updateGithubUsername = setGithubUsername;
13
14  function updateHighestCount(count) {
15    if (count > highestCount) {
16      setHighestCount(count);
17    }
18  }
19
20  return (
21    <UserContext.Provider
22      value={{
23        name,
24        githubUsername,
25        highestCount,
26        updateName,
27        updateGithubUsername,
28        updateHighestCount,
29      }}
30    >
31      { children }
32    </UserContext.Provider>
33  )
34 }
35
36 const useUser = () => useContext(UserContext);
37
38 export default useUser;
39
```

```
1 import { createSlice } from '@reduxjs/toolkit';
2
3 const initialState = {
4   name: 'Pessoa Anônima',
5   githubUsername: null,
6   highestCount: 0,
7 };
8
9 const reducers = {
10   updateName: (state, { payload }) => {
11     state.name = payload;
12   },
13
14   updateGithubUsername: (state, { payload }) => {
15     state.githubUsername = payload;
16   },
17
18   updateHighestCount: (state, { payload }) => {
19     if (payload > state.highestCount) {
20       state.highestCount = payload;
21     }
22   },
23 }
24
25 export const userSlice = createSlice({
26   name: 'user',
27   initialState,
28   reducers,
29 });
30
31 export const { updateName, updateGithubUsername, updateHighestCount } = userSlice.actions;
32
33 export default userSlice.reducer;
34
```

# Provendo o Contexto

useContext

index

```
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import './index.css';
4 import App from './App';
5 import { UserProvider } from './hooks/useUser';
6 import { CounterProvider } from './hooks/useCounter';
7
8 /* Nota: Context ao redor de toda a aplicação
9    geralmente é uma má ideia.
10   Na vida real, precisaríamos pensar em como
11   colocar cada contexto ao redor de cada grupo
12   de componentes... */
13
14 ReactDOM.render(
15   <React.StrictMode>
16     <UserProvider>
17       <CounterProvider>
18         <App />
19       </CounterProvider>
20     </UserProvider>
21   </React.StrictMode>,
22   document.getElementById('root')
23 );
24
```

RTK

index

```
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import './index.css';
4 import App from './App';
5 import { store } from './store';
6 import { Provider } from 'react-redux';
7
8 ReactDOM.render(
9   <React.StrictMode>
10     <Provider store={ store }>
11       <App />
12     </Provider>
13   </React.StrictMode>,
14   document.getElementById('root')
15 );
16
```

store

```
1 import { configureStore } from "@reduxjs/toolkit";
2 import counter from './slices/counter';
3 import user from './slices/user';
4
5 export const store = configureStore({
6   reducer: {
7     counter,
8     user,
9   },
10 });
11
```



# Consumindo o Contexto - Layout

## useContext

src/components/Layout

```
1 import React from 'react';
2 import useUser from '../hooks/useUser';
3
4 function Layout({ children }) {
5   const user = useUser();
6
7   return(
8     <>
9       <header className="body-header">
10        <h1>Contador Simprão</h1>
11        <div>
12          <img src={`https://github.com/${user.githubUsername}.png`} alt={`Avatar for github user ${user.githubUsername}`} />
13          <section>
14            <h1>Olá, <strong>{ user.name }</strong></h1>
15            <p>0 máximo que colocou o contador foi: <strong>{ user.highestCount }</strong></p>
16          </section>
17        </div>
18      </header>
19      { children }
20      <footer>Exemplo simples de Redux Toolkit.</footer>
21    </>
22  );
23 }
24
25 export default Layout;
26
```

## RTK

src/components/Layout

```
1 import React from 'react';
2 import { useSelector } from 'react-redux';
3
4 function Layout({ children }) {
5   const user = useSelector((state) => state.user);
6
7   return(
8     <>
9       <header className="body-header">
10        <h1>Contador Simprão</h1>
11        <div>
12          <img src={`https://github.com/${user.githubUsername}.png`} alt={`Avatar for github user ${user.githubUsername}`} />
13          <section>
14            <h1>Olá, <strong>{ user.name }</strong></h1>
15            <p>0 máximo que colocou o contador foi: <strong>{ user.highestCount }</strong></p>
16          </section>
17        </div>
18      </header>
19      { children }
20      <footer>Exemplo simples de Redux Toolkit.</footer>
21    </>
22  );
23 }
24
25 export default Layout;
26
```

# Consumindo o Contexto - App

useContext

App

```
1 import { useEffect, useState } from 'react';
2 import useCounter from './hooks/useCounter';
3 import useUser from './hooks/useUser';
4 import Layout from './components/Layout';
5
6 function App() {
7   const { updateName, updateGithubUsername, updateHighestCount } = useUser();
8   const { count, increment, decrement } = useCounter();
9
10  const [name, setName] = useState('');
11  const [githubUsername, setGithubUsername] = useState('');
12
13  useEffect(() => {
14    updateHighestCount(count);
15  }, [count, updateHighestCount]);
16
17  function handleSubmit(e) {
18    e.preventDefault();
19
20    const { id } = e.target;
21
22    switch (id) {
23      case 'name-form':
24        updateName(name);
25        break;
26      case 'github-username-form':
27        updateGithubUsername(githubUsername);
28        break;
29      default:
30        break;
31    }
32  }
33
34  return (
35    <Layout>
36      <main className="App">
37        <p>0 valor atual é: <strong>{ count }</strong></p>
38
39        <section className="counter-controls-section" aria-label="counter-controls">
40          <div>
41            <button onClick={ () => increment() }>Aumentar</button>
42            <button onClick={ () => decrement() }>Diminuir</button>
43          </div>
44        </section>
45
46        <section arial-label="user-controls">
47          <form id="name-form" onSubmit={ handleSubmit }>
48            <label htmlFor="name-update">
49              Nome
50              <input
51                id="name-update"
52                type="text"
53                value={ name }
54                onChange={ ({ target }) => setName(target.value) }
55              />
56            </label>
57
58            <button type="submit">Atualizar meu nome</button>
59          </form>
60
61          <form id="github-username-form" onSubmit={ handleSubmit }>
62            <label htmlFor="name-update">
63              Username do GitHub
64              <input
65                id="name-update"
66                type="text"
67                value={ githubUsername }
68                onChange={ ({ target }) => setGithubUsername(target.value) }
69              />
70            </label>
71
72            <button type="submit">Atualizar meu avatar</button>
73          </form>
74        </section>
75      </main>
76    </Layout>
77  );
78 }
79
80 export default App;
81
```

RTK

App

```
1 import { useEffect, useState } from 'react';
2 import { useSelector, useDispatch } from 'react-redux';
3 import { updateHighestCount, updateName, updateGithubUsername } from './slices/user';
4 import { increment, decrement } from './slices/counter';
5 import Layout from './components/Layout';
6
7 function App() {
8   const count = useSelector((state) => state.counter.value);
9   const dispatch = useDispatch();
10  const [name, setName] = useState('');
11  const [githubUsername, setGithubUsername] = useState('');
12
13  useEffect(() => {
14    dispatch(updateHighestCount(count));
15  }, [dispatch, count]);
16
17  function handleSubmit(e) {
18    e.preventDefault();
19
20    const { id } = e.target;
21
22    switch (id) {
23      case 'name-form':
24        dispatch(updateName(name));
25        break;
26      case 'github-username-form':
27        dispatch(updateGithubUsername(githubUsername));
28        break;
29      default:
30        break;
31    }
32  }
33
34  return (
35    <Layout>
36      <main className="App">
37        <p>0 valor atual é: <strong>{ count }</strong></p>
38
39        <section className="counter-controls-section" aria-label="counter-controls">
40          <div>
41            <button onClick={ () => dispatch(increment()) }>Aumentar</button>
42            <button onClick={ () => dispatch(decrement()) }>Diminuir</button>
43          </div>
44        </section>
45
46        <section arial-label="user-controls">
47          <form id="name-form" onSubmit={ handleSubmit }>
48            <label htmlFor="name-update">
49              Nome
50              <input
51                id="name-update"
52                type="text"
53                value={ name }
54                onChange={ ({ target }) => setName(target.value) }
55              />
56            </label>
57
58            <button type="submit">Atualizar meu nome</button>
59          </form>
60
61          <form id="github-username-form" onSubmit={ handleSubmit }>
62            <label htmlFor="name-update">
63              Username do GitHub
64              <input
65                id="name-update"
66                type="text"
67                value={ githubUsername }
68                onChange={ ({ target }) => setGithubUsername(target.value) }
69              />
70            </label>
71
72            <button type="submit">Atualizar meu avatar</button>
73          </form>
74        </section>
75      </main>
76    </Layout>
77  );
78 }
79
80 export default App;
81
```