



React Hooks

Prof. Victor Farias

v 1.1

Estado do componente

Estado do componente

- Em um componente de botão podemos querer armazenar o estado do botão
 - Quantas vezes ele foi clicado
 - Se ele está desativado
 - Se ele já foi clicado
- Usaremos **react hooks!**
 - **useState**
- Usaremos, principalmente, para armazenar dados para lógica interna do componente (lógica de apresentação)
- Eventualmente, usaremos para guardar dados de alunos, matrículas, posts, usuários ...
 - Geralmente, isso não se faz. Mas faremos para facilitar um pouco a arquitetura da aplicação

useState()

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

```
function Example() {  
  const [count, setCount] = useState(0);  
}
```

- useState recebe o valor inicial do estado
 - Nesse caso, o contador começa em 0
- useState retorna dois valores
 - Primeiro valor é o estado atual (count)
 - Segundo valor é um função para atualizar o estado atual (setCount)

Efeitos Colaterais

Efeitos colaterais

- Em um componente, queremos fazer ações demoradas:
 - Requisições
 - Processamentos pesados
 - Efeitos colaterais
- Mas, fazer antes do componente renderizar, pode demorar a renderização
- Usaremos **react hooks!**
 - **useEffect**
- useEffect recebe uma função a ser executada **depois** do componente renderizar
- Normalmente, é utilizado para fazer requisições assíncronas que demoraram, como requisições para a API REST

Exemplo

```
function Example() {  
  const [count, setCount] = useState(0);  
  
  useEffect(() => {  
    document.title = `Você clicou ${count} vezes`;  
  }, []);  
}
```

Formulários

Formulários

- Como capturar as informações de um formulário?
- **React Hook Forms**
- Hook **useForm**
 - Recebe nada
 - Retonar dois valores:
 - Primeiro é um objeto para registrar os campos
 - Segundo é o que executa a ação ao submeter o form
- Instalação:
 - `npm install react-hook-form`

Exemplo

```
import { useForm } from "react-hook-form";
export default function Example() {
  const { register, handleSubmit } = useForm();
  const onSubmit = (data) => console.log(data);
  return (
    <form onSubmit={handleSubmit(onSubmit)}>
      <input {...register("name")} />
      <select {...register("gender")} >
        <option value="female">female</option>
        <option value="male">male</option>
        <option value="other">other</option>
      </select>
      <input type="submit" />
    </form>
  );
}
```

Estado Global



Estado Global

- Em várias partes da aplicação, queremos armazenar dados que serão usadas na aplicação inteira
 - Token de autenticação
 - Informações do usuário
- Como acessar esses dados na aplicação inteira?
- Hook **useContext**
- No nosso sistema, colocaremos nossos dados de aplicação no componente App e usaremos useContext para distribuir esse dado para a aplicação toda
 - Isso não é o mais recomendado. Em aplicações reais, vale a pena usar um arquitetura como o Redux. Mas não vamos abordar isso nesse disciplina.

Como usar?

1. Crie um contexto. O contexto é onde vamos armazenar nossa variáveis

// App.js

```
import { createContext } from 'react';
```

```
export const AuthContext = createContext(null);
```

2. Envolve os componentes que precisam ter acesso às variáveis pelo contexto.
Aqui, vamos envolver a aplicação inteira.

// App.js

```
<AuthContext.Provider value={{token:"123"}}>
```

// todos os componentes em APP

```
</AuthContext.Provider>
```

Como usar?

3. Agora, qualquer componente pode usar useContext para receber o token

```
// PaginaLogin.js
```

```
import { useContext } from "react";  
import { AuthContext } from "../../App";
```

```
const user = useContext(AuthContext);  
console.log(user);
```

Como usar?

4. Para manipular o token, vamos colocar o token como estado do componente App e passar o token e setToken pelo contexto para que todos consigam acessar e modificar o token

```
// App.js
```

```
const [token, setToken] = useState(null)
```

```
<TokenContext.Provider value={{token:token, setToken:setToken}}>
```

```
  // todos os componentes em APP
```

```
</AuthContext.Provider>
```

Perguntas?

Prof. Victor Farias