



React

Não faça
chamadas a API
SEM Abort no
useEffect!

20 = R\$ 20,00

@avily.re

O problema

É muito comum vermos em aplicações React requisições sendo feitas dentro do hook `useEffect`, que nos permite fazer requisições e popular nosso componente com esses dados de forma mais dinâmica. A princípio esse não é o problema.

A grande dificuldade dessa prática é: Se por algum motivo o componente o qual está fazendo essa requisição for desmontado, como fica essa requisição em andamento ? já que o componente não está mais sendo utilizado ?

Exemplos

O que vai acontecer é que o seu componente pode causar um grande vazamento de memória levando a perda de performance na sua aplicação. Vamos ver uma exemplo do problema e de como fazer de forma incorreta:

```
export function Component() {  
  useEffect(() => {  
    axios.get("<api-url>").then(function (resolve) {  
      console.log(resolve);  
    });  
  }, []);  
}
```


Agora vamos ver como corrigir esse problema utilizando o objeto `AbortController` do próprio Javascript:

```
export function Component() {  
  useEffect(() => {  
    const controller = new AbortController();  
  
    axios.get("<api-url>", {  
      signal: controller.signal  
    }).then(function (resolve) {  
      console.log(resolve);  
    });  
  
    return () => {  
      controller.abort();  
    }  
  }, []);  
}
```

O que foi feito ?

1. Criamos um novo objeto utilizando o `AbortController` e guardamos numa constante chamada `controller`
2. Passamos o sinal (`signal`) desse `controller` para o `axios` ficar conectado a ele, dessa forma podemos controlar quando a requisição deve ser interrompida
3. definimos um `return` com uma função que é executada quando o componente é desmontado
4. Dentro desse `return` usamos o `controller` para mandar o sinal de abortar (`abort`) a requisição.

Gostou da dica ?

Compartilhe com mais pessoas!



Curta



Salve



Compartilhe

@avily.re