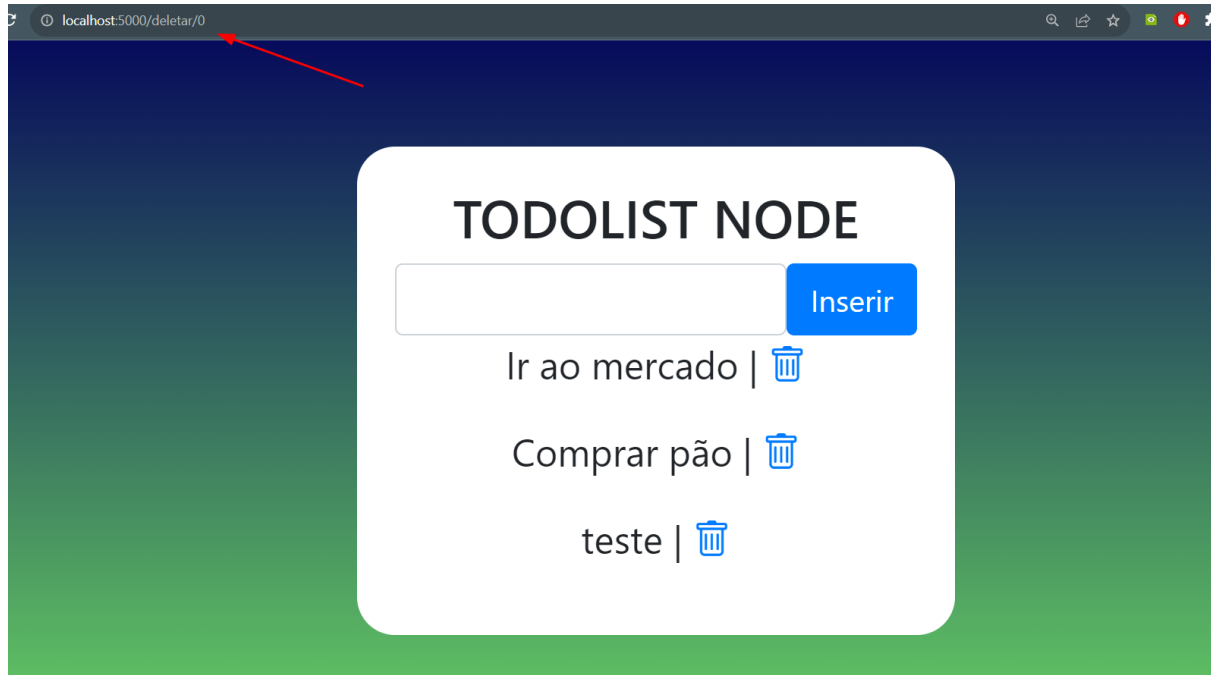


Conteúdo

Módulo 1: Projeto TodoList - Início

- Visão geral do projeto finalizado
- Bug da rota do Deletar
- Refatoração do código

Situação problema bug do deletar, quando deletamos um produto da lista a rota do produto permanece e ocasiona um **delete** infinito até limpar o array e dar erro na tela.



Atividade

Quero que tentem resolver, e a tarefa de vocês nessa OT é de me enviar o código de como resolveram, ao final irei ensinar a forma que encontrei para isso, mas existem várias formas ok?

Mãos à obra então?

Resolvendo o problema:

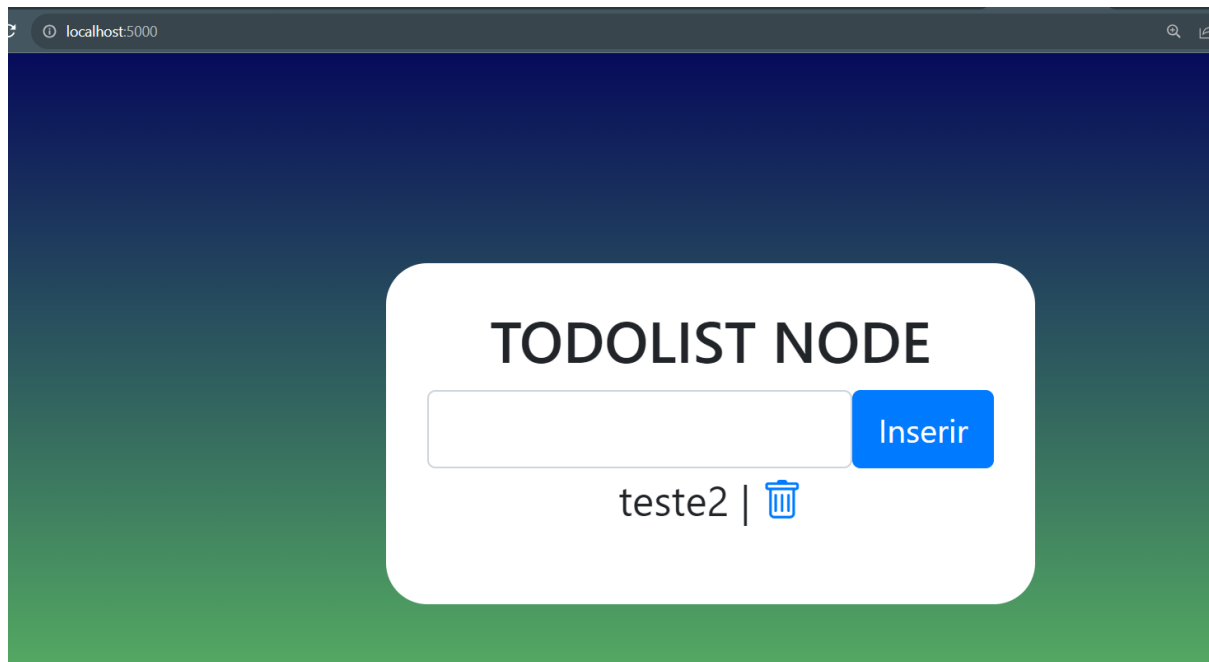
O problema que você está enfrentando parece ser que, ao deletar um produto da lista na rota `/deletar/:id`, o ID do produto e sua posição no array não são ajustados. Isso pode causar inconsistências na exibição dos produtos.

Para redirecionar o usuário de volta para a rota `'/'` após a exclusão de um item da lista, você pode usar o método `res.redirect()` do Express. Isso direciona o usuário de volta para a rota principal após a exclusão do item da lista.

Aqui está o ajuste na sua rota `/deletar/:id` para incluir o redirecionamento:

```
1 app.get('/deletar/:id',(req,res)=>{
2   const id = parseInt(req.params.id);
3   if (id >= 0 && id < tasks.length) {
4     tasks.splice(id, 1);
5     // Remove o item do array na posição do ID
6   }
7   res.redirect('/');
8   // Redireciona de volta para a rota principal '/'
9 });
```

Agora, após a exclusão do item da lista na rota `/deletar/:id`, o código irá redirecionar automaticamente o usuário de volta para a rota principal, que renderiza a página inicial com a lista atualizada de tarefas. Isso deve resolver o problema de não redirecionamento após a exclusão.



Vamos melhorar mais a nossa aplicação?

Vamos mudar o **alert** simples para um **alert** mais bonito? Vamos usar uma biblioteca externa gratuita chamada **SweetAlert2** para deixar com uma cara mais bonita?

link da biblioteca:

<https://sweetalert2.github.io/>

Para substituir o **alert** padrão por um **alert** mais estilizado usando a biblioteca **SweetAlert2**, você precisa fazer algumas alterações no seu código JavaScript. Primeiro, certifique-se de incluir a biblioteca **SweetAlert2** no seu projeto, seja por meio de um CDN ou instalado localmente.

Aqui está um exemplo de como você pode substituir o alert simples pelo **SweetAlert2**:

Inclua a biblioteca **SweetAlert2** no seu HTML. Certifique-se de fazer isso antes de usar a função `swal`.

```
1 <!-- Adicione o SweetAlert2 CSS e JS (pode ser via CDN ou baixando localm  
   ente) -->  
2 <link rel="stylesheet" href=  
   "https://cdn.jsdelivr.net/npm/sweetalert2@11/sweetalert2.min.css">  
3 <script src=  
   "https://cdn.jsdelivr.net/npm/sweetalert2@11/sweetalert2.all.min.js"></  
   script>
```

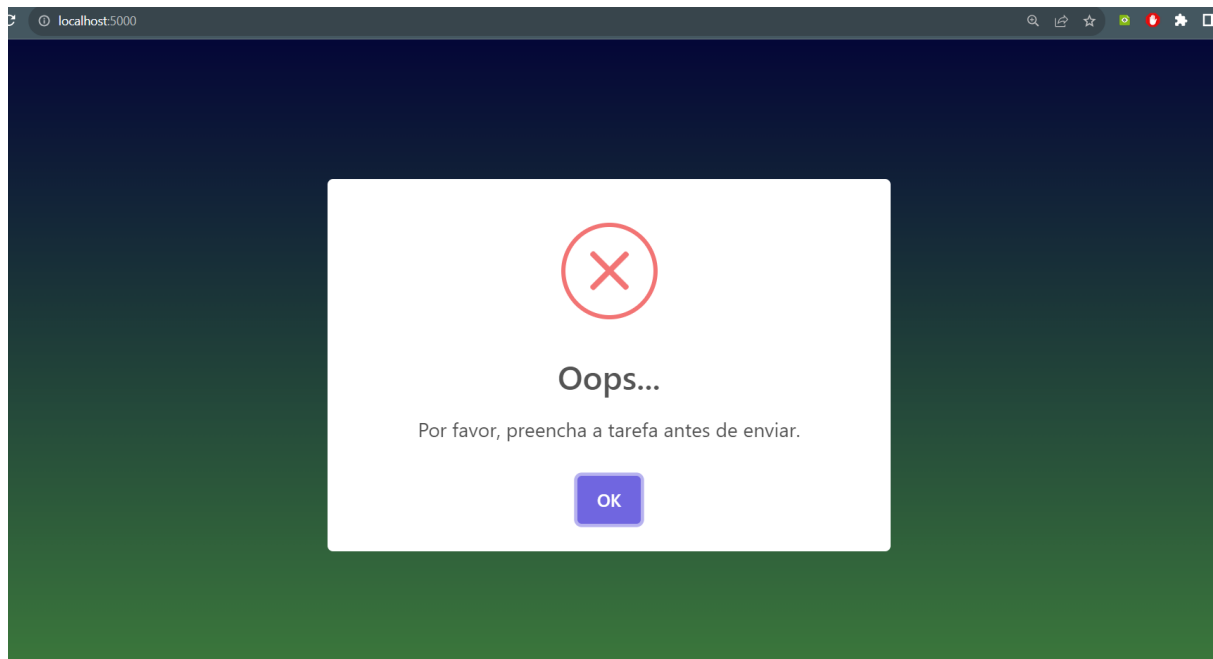
Lembrem-se um é link e fica acima no css e o script fica abaixo da tag body.

Substitua a função `alert` por `Swal.fire` no seu código JavaScript:

```
1 <script>  
2   function validarFormulario() {  
3     var tarefa = document.getElementsByName("task")[0].value;  
4     if (tarefa.trim() === '') {  
5       Swal.fire({  
6         icon: 'error',  
7         title: 'Oops...',  
8         text: 'Por favor, preencha a tarefa antes de enviar.',  
9       });  
10      return false; // Evita o envio do formulário se o campo estiver vazio  
11    }  
12    return true; // Permite o envio do formulário se o campo estiver preenchido  
13  }  
14 </script>
```

e o resultado será:

UnisenAI



Finalizamos nossas OTs de **NodeJs**, **parabéns** para todos que chegaram até aqui.