

Aplicações para a Internet II

Node.js + express + React.js + PostGres

2021/2022

<Apagar (backend)/>

Nesta secção é implementado a remoção de **employee's** da base de dados.

Node.js (backend)

No backend é necessário adicionar a rota **delete** e adicionar ao controlador do **employee** de forma a remover na base de dados quando acedemos a **/employee/delete/userId**.

src/routes/employeeRoute.js

Alterar o ficheiro de forma a incluir a rota para o **/delete**:

```
router.post('/delete', employeeController.delete);
```

src/routes/employeeController.js

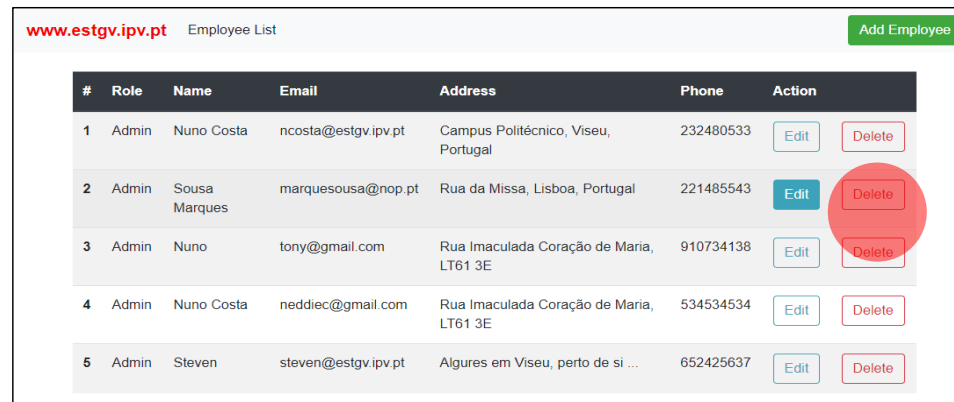
Adicionar o código:

```
controllers.delete = async (req, res) => {  
  // parâmetros por post  
  const { id } = req.body;  
  // delete por sequelize  
  const del = await Employee.destroy({  
    where: { id: id }  
  })  
  res.json({ success: true, deleted: del, message: "Deleted successful" });  
}
```

<Apagar (frontend)/>

React.js (frontend)

No **frontend** apenas necessitamos de alterar a lista de **employee**, de forma a que quando clicarmos no botão “Delete” seja chamada a API do nosso **backend** para remover os dados da base de dados.



#	Role	Name	Email	Address	Phone	Action
1	Admin	Nuno Costa	ncosta@estgv.ipv.pt	Campus Politécnico, Viseu, Portugal	232480533	<button>Edit</button> <button>Delete</button>
2	Admin	Sousa Marques	marquesousa@nop.pt	Rua da Missa, Lisboa, Portugal	221485543	<button>Edit</button> <button>Delete</button>
3	Admin	Nuno	tony@gmail.com	Rua Imaculada Coração de Maria, LT61 3E	910734138	<button>Edit</button> <button>Delete</button>
4	Admin	Nuno Costa	neddiec@gmail.com	Rua Imaculada Coração de Maria, LT61 3E	534534534	<button>Edit</button> <button>Delete</button>
5	Admin	Steven	steven@estgv.ipv.pt	Algures em Viseu, perto de si ...	652425637	<button>Edit</button> <button>Delete</button>

Quando se remove algo, deve existir uma confirmação de forma que seja possível cancelar o pedido, para isso vamos instalar uma livreria de criar alertas, o **Sweetalert** (mais informações em <https://github.com/sweetalert2/sweetalert2>). Para isso precisamos de executar o seguinte comando:

```
npm install --save sweetalert2
```

```
C:\nodejsSites\ai2_ex2\frontend>npm install --save sweetalert2
npm WARN tsutils@3.17.1 requires a peer of typescript@>=2.8.0 || >= 3.2.0-dev || >= 3.3.0-dev || >= 3.4.0-dev || >= 3.5.0-dev || >= 3.6.0-dev || >= 3.6.0-beta || >= 3.7.0-dev || >= 3.7.0-beta but none is installed. You must install peer dependencies yourself.

+ sweetalert2@9.7.0
added 1 package from 6 contributors and audited 911062 packages in 12.515s

39 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

<Apagar (frontend)/>

Sweetalert (mais informações em <https://github.com/sweetalert2/sweetalert2>). Também é necessário instalar o SASS (Linguagem de extensão ao CSS). Para isso, usamos o comando:

```
npm install node-sass
```

```
C:\nodejsSites\ai2_ex2\frontend>npm install node-sass

> node-sass@4.13.1 install C:\nodejsSites\ai2_ex2\frontend\node_modules\node-sass
> node scripts/install.js

Downloading binary from https://github.com/sass/node-sass/releases/download/v4.13.1/win32-x64-72_binding.node
Download complete.
Binary saved to C:\nodejsSites\ai2_ex2\frontend\node_modules\node-sass\vendor\win32-x64-72\binding.node
Caching binary to C:\Users\ncost\AppData\Roaming\npm-cache\node-sass\4.13.1\win32-x64-72_binding.node

> node-sass@4.13.1 postinstall C:\nodejsSites\ai2_ex2\frontend\node_modules\node-sass
> node scripts/build.js

Binary found at C:\nodejsSites\ai2_ex2\frontend\node_modules\node-sass\vendor\win32-x64-72\binding.node
Testing binary
Binary is fine
npm WARN tsutils@3.17.1 requires a peer of typescript@>=2.8.0 || >= 3.2.0-dev || >= 3.3.0-dev || >= 3.4.0-dev || >= 3.5.0-dev || >= 3.6.0-dev || >= 3.6.0-beta || >= 3.7.0-dev || >= 3.7.0-beta but none is installed. You must install peer dependencies yourself.

+ node-sass@4.13.1
added 98 packages from 54 contributors and audited 911587 packages in 14.082s

39 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

<Apagar (frontend)/>

src/view/list.js

Neste ficheiro é necessário incluir a importação da nova livreria e alterar o botão de remover ficando o seguinte código:

```
//sweetalert2 - importação
import Swal from 'sweetalert2/dist/sweetalert2.js'
import 'sweetalert2/src/sweetalert2.scss'
. . .
<button class="btn btn-outline-danger" onClick={()=>onDelete(data.id)}> Delete </button>
. . .
```

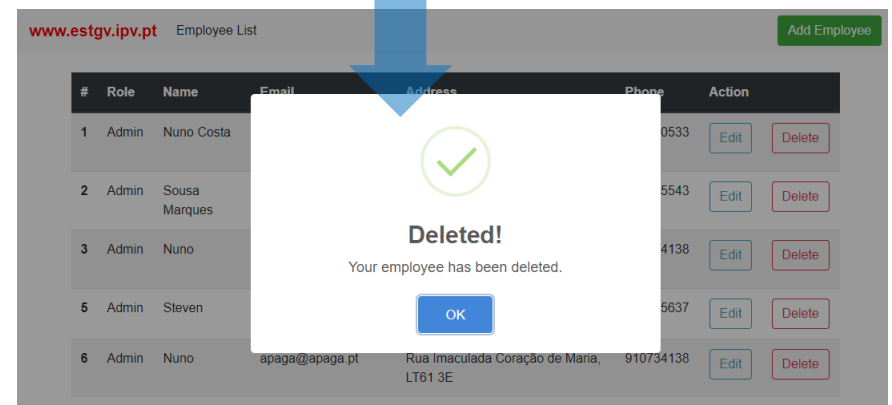
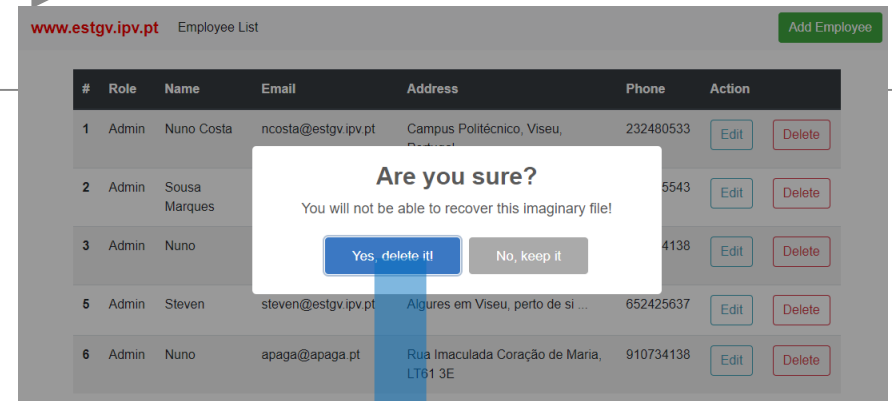
Também é necessário acrescentar a função onDelete, usando o Sweetalert:

```
onDelete(id) {
  Swal.fire({
    title: 'Are you sure?',
    text: 'You will not be able to recover this
imaginary file!',
    type: 'warning',
    showCancelButton: true,
    confirmButtonText: 'Yes, delete it!',
    cancelButtonText: 'No, keep it'
  }).then((result) => {
    if (result.value) {
      onDelete(id)
    } else if (result.dismiss ===
Swal.DismissReason.cancel) {
      Swal.fire(
        'Cancelled',
        'Your imaginary file is safe :)',
        'error'
      )
    }
  })
}
```

<Apagar (frontend)/>

Caso se confirme o apagar, é executada a função **sendDelete** criada no mesmo ficheiro:

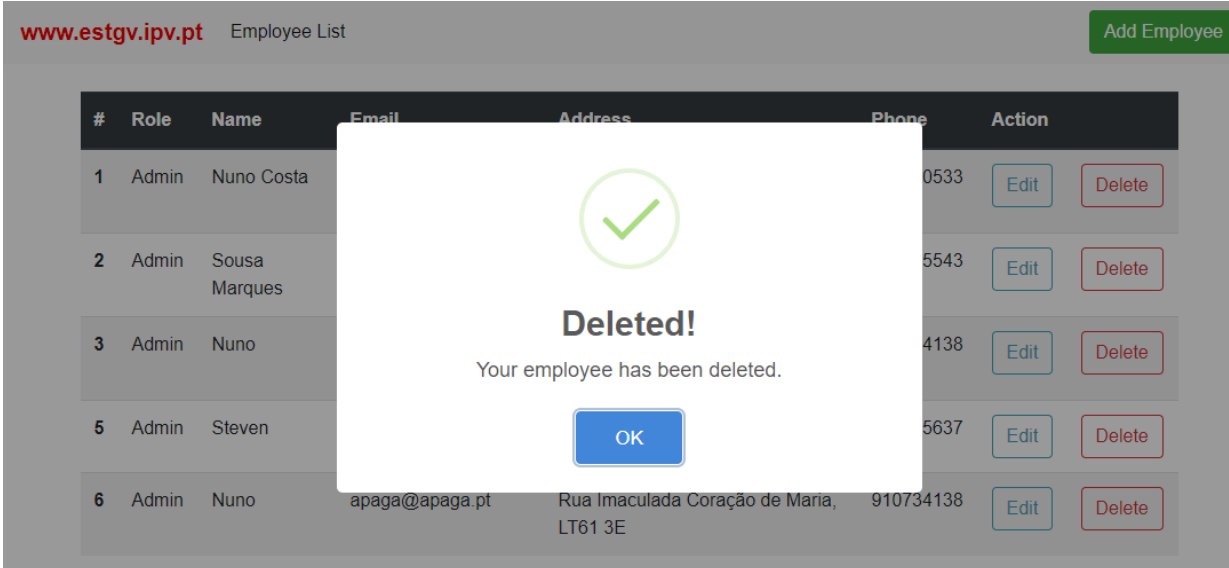
```
SendDelete(userId)
{
  // url do backend
  const baseUrl = "http://localhost:3000/employee/delete"
  // network
  axios.post(baseUrl, {
    id:userId
  })
  .then(response =>{
    if (response.data.success) {
      Swal.fire(
        'Deleted!',
        'Your employee has been deleted.',
        'success'
      )
    }
  })
  .catch ( error => {
    alert("Error 325 ")
  })
}
```



<Apagar (frontend)/>

PROBLEMA

Ao apagar um **employee**, e após a confirmação para apagar, verifica-se que a lista não é atualizada. Como podemos resolver o problema?



The screenshot shows a web application interface for managing employees. At the top left is the URL 'www.estgv.ipv.pt' and the title 'Employee List'. At the top right is a green 'Add Employee' button. Below is a table with columns: #, Role, Name, Email, Address, Phone, and Action. The table contains six rows of employee data. A white modal dialog box is centered over the table, displaying a green checkmark icon, the text 'Deleted!', and 'Your employee has been deleted.' with an 'OK' button.

#	Role	Name	Email	Address	Phone	Action
1	Admin	Nuno Costa			0533	<button>Edit</button> <button>Delete</button>
2	Admin	Sousa Marques			5543	<button>Edit</button> <button>Delete</button>
3	Admin	Nuno			4138	<button>Edit</button> <button>Delete</button>
5	Admin	Steven			5637	<button>Edit</button> <button>Delete</button>
6	Admin	Nuno	apaga@apaga.pt	Rua Imaculada Coração de Maria, LT61 3E	910734138	<button>Edit</button> <button>Delete</button>

<Apagar (frontend)/>

SOLUÇÃO: Alterar a função **componentDidMount**, criando uma outra função **loadEmployee**, ficando o código da seguinte forma:

```
useEffect(() => {  
  
    LoadEmployee();  
  
}, []);  
  
function LoadEmployee() {  
    const url = "http://localhost:3000/employee/list";  
    axios.get(url)  
    .then(res => {  
        if(res.data.success){  
            const data = res.data.data;  
            setDataEmployee(data);  
        }else{  
            alert("Error Web Service!");  
        }  
    })  
    .catch(error => {  
        alert(error)  
    });  
}
```


<Apagar (frontend)/>

Depois é só acrescentar a função **loadEmployee** dentro da função **sendDelete** e caso a resposta do lado do **backend** for bem-sucedida. Assim teremos a função com o seguinte aspeto:

```
sendDelete(userId)
{
  // url do backend
  const baseUrl = "http://localhost:3000/employee/delete"
  // network
  axios.post(baseUrl, {
    id:userId
  })
  .then(response =>{
    if (response.data.success) {
      Swal.fire(
        'Deleted!',
        'Your employee has been deleted.',
        'success'
      )
      LoadEmployee()
    }
  })
  .catch ( error => {
    alert("Error 325 ")
  })
}
```

Bibliografia ...

- <https://devcenter.heroku.com/>
- <https://reactjs.org/docs/getting-started.html>
- <https://nodejs.org/en/docs/>
- <https://developer.mozilla.org/pt-BR/>
- https://developer.mozilla.org/pt-BR/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started
- https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/Express_Nodejs/Introduction
- <https://www.bezkoder.com/react-node-express-mysql/>
- <https://developer.okta.com/blog/2018/07/10/build-a-basic-crud-app-with-node-and-react>