

Aplicações para a Internet II

Node.js + express + React.js + PostGres

2021/2022

<Criar (backend)/>

Nesta secção pretende-se que haja a ligação entre o Node.js (**backend**) e o React.js (**frontend**) para o registo de **employee's** na base de dados.

Node.js (backend)

No **backend** é necessário alterar a rota e o controlador do **employee** de forma a registar os dados na base de dados quando acedemos a **/create**.

src/routes/employeeRoute.js

Alterar o ficheiro de forma a incluir a rota para o **/create**, assim teremos o código seguinte:

```
. . .  
router.get('/list',employeeController.list);  
router.post('/create',employeeController.create);  
  
module.exports = router;
```

<Criar (backend) />

src/controllers/employeeController.js

Acrescentar no controlador a tarefa de criar (**create**), adicionando o código seguinte:

```
/* REGISTRAR ----- */
controllers.create = async (req,res) => {
  // data
  const { name, email, address, phone, role
} = req.body;
  // create
  const data = await Employee.create({
    name: name,
    email: email,
    address: address,
    phone: phone,
    roleId: role
  })
  .then(function(data){
    return data;
  })
  .catch(error =>{
    console.log("Erro: "+error)
    return error;
  })
  // return res
  res.status(200).json({
    success: true,
    message:"Registado",
    data: data
  });
}
```

<React.js – Criar (frontend)/>

No frontend é necessário alterar o módulo do formulário para registrar o employee.

src/view/form.js

Sendo assim, o ficheiro terá que ter, a inicialização dos **dados a enviar**, o formulário a **preencher** aquando do registo, a **validação dos dados** de entrada e **chamar a API create** localizada em **<http://localhost:3000/employee/create>**.

```
import 'bootstrap/dist/css/bootstrap.min.css';
import 'bootstrap/dist/js/bootstrap.bundle.min';
import axios from 'axios';
import React, {useEffect, useState} from "react";

export default function EditComponent(){

  const [dataEmployee, setdataEmployee] = useState("");
  const [campName, setcampName] = useState("");
  const [campEmail, setcampEmail] = useState("");
  const [campPhone, setcampPhone] = useState("");
  const [campAddress, setcampAddress] = useState("");
  const [stringRole, setstringRole] = useState("");
  const [selectRole, setselectRole] = useState("");

  . . .
```

<Criar (frontend) />

```

    ...
    return (
      <div>
        <div className="form-row justify-content-center">
          <div className="form-group col-md-6">
            <label htmlFor="inputPassword4">Name </label>
            <input type="text" className="form-control"
placeholder="Name"
value={campName} onChange={value=>
setcampName(value.target.value)} />
          </div>
          <div className="form-group col-md-6">
            <label htmlFor="inputEmail4">Email</label>
            <input type="email" className="form-control"
placeholder="Email"
value={campEmail} onChange={value=>
setcampEmail(value.target.value)} />
          </div>
        </div>
        <div className="form-row">
          <div className="form-group col-md-6">
            <label htmlFor="inputState">Role</label>
            <select id="inputState" className="form-control"
onChange={value=> setselectRole(value.target.value)}>
              <option defaultValue>Choose...</option>
              <option value="1">Admin...</option>
              <option value="2">Project Manager</option>
              <option value="3">Programer</option>
            </select>
          </div>
          <div className="form-group col-md-6">
            <label htmlFor="inputEmail4">Phone</label>
            <input type="number" className="form-control"
placeholder="Phone" value={campPhone}
onChange={value=> setcampPhone(value.target.value)} />
          </div>
        </div>
        <div className="form-group">
          <label htmlFor="inputAddress">Address</label>
          <input type="text" className="form-control"
id="inputAddress" placeholder="1234 Main St"
value={campAddress} onChange={value=>
setcampAddress(value.target.value)} />
        </div>
        <button type="submit" className="btn btn-primary"
onClick={()=>SendSave()}>Save</button>
      </div>
    );
  };
}

```

```

    </div>
  );
};
}

function SendSave(){
  if (selectRole===0) {
    alert("Choose Role!")
  }
  else if (campPhone=== "") {
    alert("Insert the Phone!")
  }
  else if (campName=== "") {
    alert("Insert Name!")
  }
  else if (campEmail=== "") {
    alert("Insert Email!")
  }
  else if (campAddress=== "") {
    alert("Insert Address!")
  }
  else {
    const baseUrl = "http://localhost:3000/employee/create"
    const datapost = {
      name : campName,
      email : campEmail,
      phone : campPhone,
      address : campAddress,
      role : selectRole
    }
    axios.post(baseUrl,datapost)
    .then(response=>{
      if (response.data.success===true) {
        alert(response.data.message)
      }
      else {
        alert(response.data.message)
      }
    }).catch(error=>{
      alert("Error 34 "+error)
    })
  }
}
}

```

<Selecionar para Editar (Backend)/>

Nesta secção pretende-se que haja a ligação entre o Node.js (**backend**) e o React.js (**frontend**) para seleccionar um dado **employee** de forma que depois (na Secção Parte 7) seja possível editar os dados num formulário e alterar na base de dados.

Node.js (backend)

No **backend** é necessário alterar a rota e o controlador do **employee** de forma a ir buscar os dados à base de dados quando acedemos a **/employee/get/userId**.

src/routes/employeeRoute.js

Alterar o ficheiro de forma a incluir a rota para o **/get/:id**, assim teremos o código seguinte:

```
. . .  
router.get('/get/:id', employeeController.get) ;  
. . .
```

<Selecionar para Editar (Backend)/>

src/controllers/employeeController.js

Acrescentar no controlador a tarefa de selecionar (get), adicionando o código seguinte:

```
/* BUSCAR para EDITAR ----- */
controllers.get = async (req,res) => {
  const { id } = req.params;
  const data = await Employee.findAll({
    where: { id: id },
    include: [ Role ]
  })
  .then(function(data) {
    return data;
  })
  .catch(error =>{
    return error;
  })
  res.json({ success: true, data: data });
}
```

Podemos verificar se está a funcionar
accedendo ao url

<http://localhost:3000/employee/get/2>, que
será visualizado em formato JSON os
dados do **employee** com id igual a 2



<Selecionar para Editar (frontend)/>

React.js (frontend)

Nesta secção vamos alterar a listagem de forma que seja possível ir para um formulário de editar após o clicar em “Update” na listagem.

www.estgv.ipv.pt Employee List Add Employee

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

Ao clicar em “Edit” será apresentado o formulário com os dados referentes ao **employee** que seleccionou

www.estgv.ipv.pt Employee List Add Employee

Name Email

Role Phone

Address

Update

<Selecionar para Editar (frontend)/>

src/view/list.js

Alterar a listagem de forma a contemplar o clique. Para isso precisamos de incluir no ficheiro a livreria react-router:

```
import { Link } from "react-router-dom";  
.  
.  
.  
// e alterar o botão para Link:  
<Link class="btn btn-outline-info " to={"/edit/"+data.id} >Edit</Link>
```

src/view/edit.js

Neste ficheiro é necessário importar a **livreria axios**, definir uma **constante com o endereço base** da nossa API (Node.js), **inicializar os dados**, ir **buscar os dados à base de dados** e criar (render) os **elementos do formulário**. O ficheiro fica com o seguinte código:

```
import 'bootstrap/dist/css/bootstrap.min.css';  
import 'bootstrap/dist/js/bootstrap.bundle.min';  
  
import axios from 'axios';  
import React ,{useEffect, useState} from "react";  
import {useParams} from "react-router-dom";  
  
const baseUrl = "http://localhost:3000";
```

<Seleccionar para Editar (frontend)/>

```
import 'bootstrap/dist/css/bootstrap.min.css';
import 'bootstrap/dist/js/bootstrap.bundle.min';

import axios from 'axios';
import React, {useEffect, useState} from 'react';
import {useParams} from 'react-router-dom';

const baseUrl = "http://localhost:3000";

export default function EditComponent(){

  const [dataEmployee, setDataEmployee] = useState("");
  const [campName, setcampName] = useState("");
  const [campEmail, setcampEmail] = useState("");
  const [campPhone, setcampPhone] = useState("");
  const [campAddress, setcampAddress] = useState("");
  const [stringRole, setstringRole] = useState("");
  const [selectRole, setselectRole] = useState("");

  const {employeeId} = useParams();

  useEffect(() => {
    const url = baseUrl+"/employee/get/" + employeeId;
    axios.get(url)
      .then(res=>{
        if (res.data.success) {
          const data = res.data.data[0];
          setDataEmployee(data);
          setcampName(data.name);
          setcampEmail(data.email);
          setcampPhone(data.phone);
          setcampAddress(data.address);
          setstringRole(data.role.role);
          setselectRole(data.roleId);
          console.log(JSON.stringify(data.role.role))
        }
        else {
          alert("Error web service")
        }
      })
      .catch(error=>{
        alert("Error server: "+error)
      })
  }, []);
```

```
return (
  <div>
    <div className="form-row justify-content-center">
      <div className="form-group col-md-6">
        <label htmlFor="inputPassword4">Name</label>
        <input type="text" className="form-control" placeholder="Name"
          value={campName} onChange={ (value) =>
            setcampName(value.target.value)} />
      </div>
      <div className="form-group col-md-6">
        <label htmlFor="inputEmail4">Email</label>
        <input type="email" className="form-control"
          placeholder="Email"
          value={campEmail} onChange={ (value) =>
            setcampEmail(value.target.value)} />
      </div>
    </div>
    <div className="form-row">
      <div className="form-group col-md-6">
        <label htmlFor="inputState">Role</label>
        <select id="inputState" className="form-control" onChange={
          (value) => setstringRole(value.target.value)}>
          <option value={selectRole}>{stringRole}</option>
          <option value="1">Admin</option>
          <option value="2">Project Manager</option>
          <option value="3">Programer</option>
        </select>
      </div>
      <div className="form-group col-md-6">
        <label htmlFor="inputEmail4">Phone</label>
        <input type="number" className="form-control"
          placeholder="Phone"
          value={campPhone} onChange={ (value) =>
            setcampPhone(value.target.value)} />
      </div>
    </div>
    <div className="form-group">
      <label htmlFor="inputAddress">Address</label>
      <input type="text" className="form-control" id="inputAddress"
        placeholder="1234 Main St"
        value={campAddress} onChange={ (value) =>
          setcampAddress(value.target.value)} />
    </div>
    <button type="submit" className="btn btn-primary">Update</button>
  </div>
);
}
```

<Alterar (backend)/>

Nesta secção pretende-se os dados alterados no formulário de edição do **employee's** seja atualizado na base de dados.

Node.js (backend)

No backend é necessário incluir a rota para o *update* e criar o controlador do **employee** para executar a alteração na base de dados quando acedemos a **/employee/update/userId**.

src/routes/employeeRoute.js

Alterar o ficheiro de forma a incluir a rota para o **/update/:id**, assim teremos o código seguinte:

```
. . .  
router.post('/update/:id', employeeController.update);  
. . .
```

<Alterar (backend)/>

src/controllers/employeeController.js

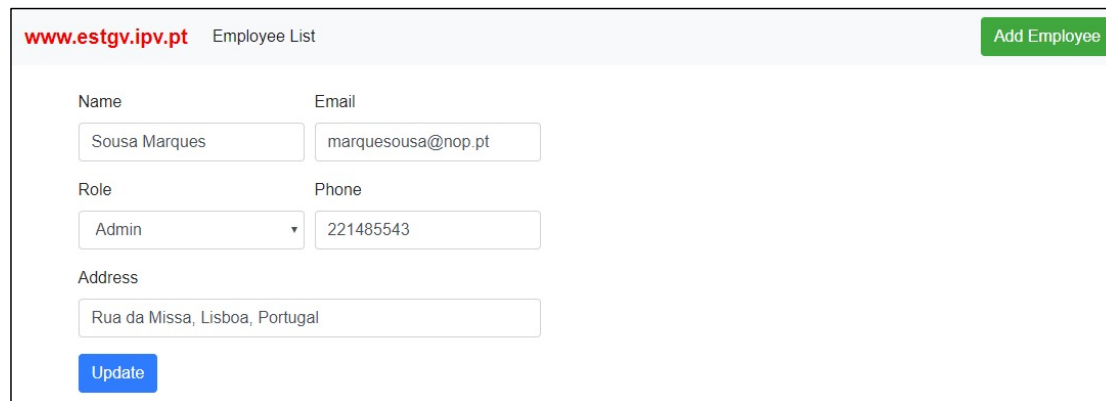
No controlador adicionar o código seguinte:

```
/* EDITAR ----- */
controllers.update = async (req,res) => {
  // parameter get id
  const { id } = req.params;
  // parameter POST
  const {name, email, address, phone, role } = req.body;
  // Update data
  const data = await Employee.update({
    name:name,
    email:email,
    address:address,
    phone:phone,
    roleId:role
  },
  {
    where: { id: id}
  })
  .then( function(data){
    return data;
  })
  .catch(error => {
    return error;
  })
  res.json({success:true, data:data, message:"Updated successful"});
}
```

<Alterar (frontend)/>

React.js (frontend)

No frontend apenas necessitamos de alterar a edição de employee, de forma a que quando clicarmos no botão “Update” seja chamada a API do nosso backend para alterar os dados na base de dados.



The screenshot shows a web application interface for managing employees. At the top left, the URL 'www.estgv.ipv.pt' is displayed in red, followed by the title 'Employee List'. In the top right corner, there is a green button labeled 'Add Employee'. The main form area contains several input fields: 'Name' with the value 'Sousa Marques', 'Email' with 'marquesousa@nop.pt', 'Role' with a dropdown menu showing 'Admin', 'Phone' with '221485543', and 'Address' with 'Rua da Missa, Lisboa, Portugal'. At the bottom left of the form, there is a blue button labeled 'Update'.

src/view/edit.js

Neste ficheiro é necessário alterar o botão de edição para:

```
<button type="submit" class="btn btn-primary" onClick={ () => this.sendUpdate() }>Update</button>  
. . .
```

<Alterar (frontend)/>

src/view/edit.js

No mesmo ficheiro é necessário a seguinte função:

```
function SendUpdate(){

    // url de backend
    const url = baseUrl+"/employee/update/"+employeeId
    const datapost = {
        name : campName,
        email : campEmail,
        phone : campPhone,
        address : campAddress,
        role : selectRole
    }

    axios.post(url,datapost)
    .then(response=>{
        if (response.data.success===true) {
            alert(response.data.message)
        }
        else {
            alert("Error")
        }
    }).catch(error=>{
        alert("Error 34 "+error)
    })
}
```