

Exercício 3

Nota 1

Como este não é um projeto front-end e não estamos a trabalhar com React, a aplicação não é criada com `create vite@latest --template react`. Inicializa-se este projeto com o comando `npm init`.

Nota 2

Como o "node_modules" é criado utilizando o "npm init", não será eliminado quando tentar adicionar o seu código ao git utilizando o "git add.", por isso crie um ficheiro chamado ".gitignore" e escreva "node_modules" para que o git o ignore sempre que tentar adicionar, confirmar ou enviar para um repositório remoto.

Forte recomendação: quando estiver a trabalhar em código de backend, esteja sempre atento ao que está a acontecer no terminal que está a executar a sua aplicação.

3.1 Passo 1 do backend da lista telefónica

Implemente uma aplicação Node que devolva uma lista codificada de entradas da lista telefónica a partir do endereço <http://localhost:3001/api/persons>.

Dados:

```
[{"id": "1", "name": "Arto Hellas", "number": "040-123456"}, {"id": "2", "name": "Ada Lovelace", "number": "39-44-5323523"}, {"id": "3", "name": "Dan Abramov", "number": "12-43-234345"}, {"id": "4", "name": "Mary Poppendieck", "number": "39-23-6423122"}]
```

Saída no browser após pedido GET:



```
[  
  {  
    id: "1",  
    name: "Arto Hellas",  
    number: "040-123456"  
  },  
  {  
    id: "2",  
    name: "Ada Lovelace",  
    number: "39-44-5323523"  
  },  
  {  
    id: "3",  
    name: "Dan Abramov",  
    number: "12-43-234345"  
  },  
  {  
    id: "4",  
    name: "Mary Poppendieck",  
    number: "39-23-6423122"  
  }]
```

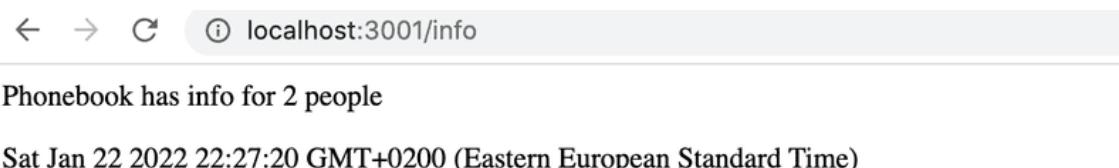
Note que a barra na rota `api/persons` não é um caracter especial e é como qualquer outro caracter da string.

A aplicação deve ser iniciada com o comando `npm start`.

A aplicação também deve oferecer um comando `npm run dev` que executará a aplicação e reiniciará o servidor sempre que forem feitas alterações e guardadas num ficheiro no código-fonte.

3.2 Passo 2 do backend da lista telefónica

Implemente uma página no endereço `http://localhost:3001/info` que se assemelhe a isto:



```
← → ⌂ ⓘ localhost:3001/info  
  
Phonebook has info for 2 people  
  
Sat Jan 22 2022 22:27:20 GMT+0200 (Eastern European Standard Time)
```

A página deve mostrar a hora a que o pedido foi recebido e quantas entradas existem na lista telefónica no momento do processamento do pedido.

3.3 Passo 3 do backend da lista telefónica

Implemente a funcionalidade para apresentar as informações de uma única entrada da lista telefónica. O URL para obter os dados de uma pessoa com o ID 5 deve ser <http://localhost:3001/api/persons/5>

Se não for encontrada uma entrada para o ID fornecido, o servidor terá de responder com o código de estado apropriado.

3.4 Passo 4 do backend da lista telefónica

Implemente uma funcionalidade que possibilite eliminar uma única entrada da lista telefónica fazendo um pedido HTTP DELETE para o URL único dessa entrada da lista telefónica.

Teste se a sua funcionalidade funciona com o Postman ou com o cliente REST do Visual Studio Code.

3.5 Passo 5 do backend da lista telefónica

Expanda o backend para que novas entradas da lista telefónica possam ser adicionadas fazendo pedidos HTTP POST para o endereço <http://localhost:3001/api/persons>.

Gere um novo ID para a entrada da agenda telefónica com a função Math.random. Utilize um intervalo suficientemente grande para os seus valores aleatórios, de modo a que a probabilidade de criar IDs duplicados seja pequena.

3.6 Passo 6 do backend da lista telefónica

Implemente o tratamento de erros para a criação de novas entradas. A solicitação não poderá ser bem-sucedida se:

- O nome ou número está em falta
- O nome já existe na lista telefónica

Responda a pedidos como estes com o código de estado apropriado e envie também informações que expliquem o motivo do erro, por exemplo:

