

Roteiro das rotas:

Rota showProductById: GET /products/:id

- Receber o id do produto por parâmetro na requisição
- Validar se é um id válido
- Buscar o produto pelo id no banco de dados, o produto retornado deve popular os dados no owner e received (se já existir)
- Verificar se o produto foi encontrado, se não foi encontrado retornar 404
- Retornar o produto para o cliente (front)

Pesquisar: req.params, isValidObjectId (do mongoose), findById (do mongoose), .populate (usado na busca do mongoose)

Rota showUserProducts: GET /products/showUserProducts

- Obter id do usuário pelo token
- Procurar os produtos pelo id do usuário referente ao campo owner
- Retornar os produtos para o cliente (front).

Pesquisar: find({"owner":id}) (do mongoose)

Rota showReceiverProducts: GET /products/showReceiverProducts

- Obter id do usuário pelo token
- Procurar os produtos pelo id do usuário referente ao campo receiver
- Retornar os produtos para o cliente (front).

Pesquisar: find({"receiver":id}) (do mongoose)

Rota updateProduct: PUT /products/:id

- Receber os dados do produto no corpo da requisição e o id do produto a ser alterado nos parâmetros da requisição
- Verificar se o id do produto enviado é válido
- Validar se todos os campos informados são válidos
- Procurar o produto
- Verificar se o produto foi encontrado, se não foi encontrado retornar 404
- Alterar o produto
- Retornar para o cliente o produto alterado (front)

Pesquisar: req.params, isValidObjectId (do mongoose), findById (do mongoose)

Rota deleteProduct: DELETE /products/:id

- Receber o id do produto por parâmetro na requisição
- Validar se é um id válido

- Buscar o produto pelo id no banco de dados
- Verificar se o produto foi encontrado, se não foi encontrado retornar 404
- Deletar o produto
- Retornar nenhum conteúdo para o cliente (front), com o status 204.

Pesquisar: req.params, isValidObjectId (do mongoose), findById (do mongoose), findByIdAndDelete (mongoose).

Rota schedule: PATCH /products/schedule/:id

- Receber o id do produto por parâmetro na requisição
- Validar se é um id válido
- Buscar o produto pelo id no banco de dados
- Verificar se o produto foi encontrado, se não foi encontrado retornar 404
- Verificar se o produto ainda está disponível, a partir do atributo available
- Obter o usuário pelo token
- Verificar se o usuário é proprietário do próprio produto que ele quer agendar (verificar se o id armazenado na propriedade owner é igual ao id do usuário resgatado pelo token) (utilize o método equals)
- Alterar o atributo receive do produto resgatado do banco de dados, armazenando o id do usuário resgatado pelo token
- Alterar o produto e retornar a mensagem: `A visita foi agendada com Sucesso, entre em contato com com \${Nome do Usuário}, pelo telefone, \${Telefone do Usuário}`

Pesquisar: req.params, isValidObjectId (do mongoose), findById (do mongoose), findByIdAndUpdate (mongoose), equals().

(Essa rota é opcional, não é obrigatório entregar) Rota concludeDonation: PATCH /products/concludeDonation/:id

- Receber o id do produto por parâmetro na requisição
- Validar se é um id válido
- Buscar o produto pelo id no banco de dados
- Verificar se o produto foi encontrado, se não foi encontrado retornar 404
- Verificar se o produto ainda está disponível, a partir do atributo available
- Obter o usuário pelo token
- Verificar se o usuário é proprietário do próprio produto que ele quer agendar (verificar se o id armazenado na propriedade owner é igual ao id do usuário resgatado pelo token) (utilize o método equals)
- Alterar o atributo available do produto resgatado da base de dados para false
- Alterar o valor do atributo donated_at com a data atual (utilizar new Date())
- Alterar o produto e retornar a mensagem: `Doação concluída com Sucesso.

Pesquisar: req.params, isValidObjectId (do mongoose), findById (do mongoose), findByIdAndUpdate (mongoose), equals(), new Date().