

Filtros, ordenação e paginação

Labenu_



O que vamos ver hoje? 🙄

- Como filtrar, ordenar e paginar dados no backend



Introdução

Labenu_



Introdução

- Imagine que você faz parte de um time contratado para arquitetar o backend de uma aplicação de receitas culinárias
- Após uma breve divisão de tarefas, as seguintes tasks já foram entregues:
 - Criar a estrutura de pastas e arquivos, com as configurações básicas do Knex e do Express
 - Criar e popular tabelas de usuários e receitas no Mysql
 - Estruturar os modelos de dados correspondentes no código
 - Criar um endpoint que retorna todas as receitas



Filtros

Labenu_



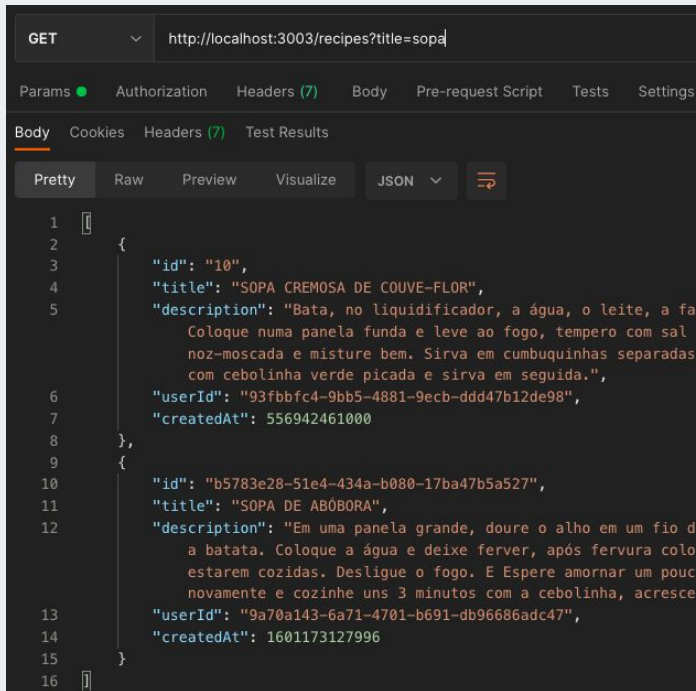
Filtros

- Pensando na escalabilidade da aplicação, o time de frontend solicita que, no endpoint já criado, as receitas possam ser **filtradas por título**.
- O líder do projeto atribui essa tarefa a você e, levando em conta sua pouca experiência, sugere um roteiro de como proceder:



Filtros

1. VISUALIZE COM CLAREZA O COMPORTAMENTO DO ENDPOINT (O QUÊ DEVE ACONTECER?)



```
GET http://localhost:3003/recipes?title=sopa

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON ↕

1 {
2   {
3     "id": "10",
4     "title": "SOPA CREMOSA DE COUVE-FLOR",
5     "description": "Bata, no liquidificador, a água, o leite, a farinha de milho e o queijo. Coloque numa panela funda e leve ao fogo, tempero com sal, pimenta-do-reino, alho e cebolinha. Sirva em conchas e decore com queijo ralado e cebolinha verde picada.",
6     "userId": "93fbbfc4-9bb5-4881-9ecb-ddd47b12de98",
7     "createdAt": 556942461000
8   },
9   {
10    "id": "b5783e28-51e4-434a-b080-17ba47b5a527",
11    "title": "SOPA DE ABÓBORA",
12    "description": "Em uma panela grande, doure o alho em um fio de óleo. Acrescente a batata, a abóbora e a cebolinha. Deixe ferver por 15 minutos. Acrescente o leite e o creme de leite. Tempere com sal e pimenta-do-reino. Sirva em conchas e decore com queijo ralado e cebolinha verde picada.",
13    "userId": "9a70a143-6a71-4701-b691-db96686adc47",
14    "createdAt": 1601173127996
15  }
16 }
```



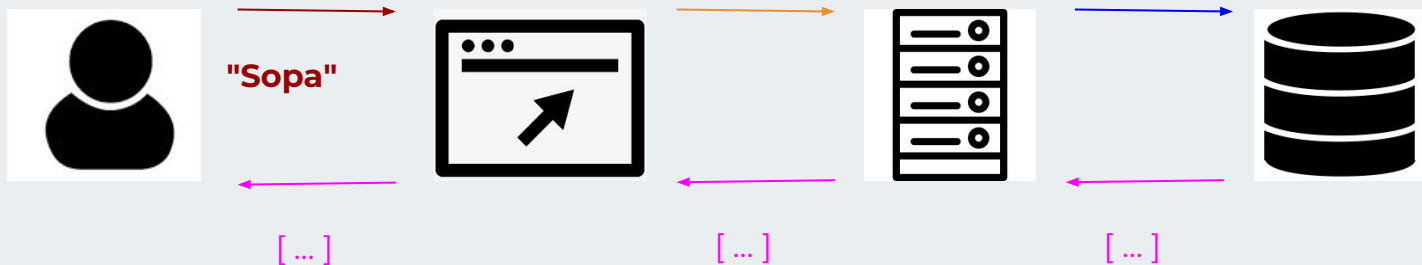
Filtros

2. ESBOCE O FLUXO DE DADOS E AS ETAPAS DO PROCESSO (COMO AS COISAS ACONTECERÃO?)

`localhost:3003/recipes?title=sopa`

```
SELECT * FROM recipes
```

```
WHERE title LIKE "%sopa%"
```



Filtros

2. ESBOCE O FLUXO DE DADOS E AS ETAPAS DO PROCESSO (COMO AS COISAS ACONTECERÃO?)

- Receber o título da receita por query string
- Consultar o banco
- Responder a requisição com o resultado da consulta



Filtros


3. ACRESCENTE O TRATAMENTO DE EXCEÇÕES (E SE ALGO DER ERRADO?)

- Receber o nome por query string
- **Validar o valor informado**
- Consultar o banco
- **Verificar se algum valor foi retornado**
- Responder a requisição com o resultado da consulta



4. PARTIR PARA A IMPLEMENTAÇÃO

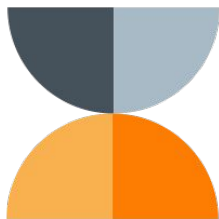


Vamos ver na prática! 



Pausa para relaxar 🤪

5 min



- Filtros podem ser feitos no front usando usando o método *filter*, mas são muito mais fáceis de fazer e mais performáticos no backend

```
SELECT * FROM ... WHERE ...
```



Ordenação

Labenu_



Ordenação

- Você fez um belo trabalho, impressionando a todos. No entanto, percebeu que algumas melhorias ainda podem ser feitas.
- Assim, você se propôs a refatorar seu endpoint seguindo o roteiro proposto por seu mentor

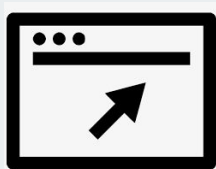


Ordenação 123

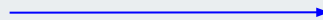
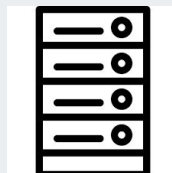
- Agora, a *query string* da requisição incluirá os parâmetros **sort**, para que os resultados possam ser ordenados por título ou data de criação, e **order**, para que a ordenação possa ser crecente ou decrecente

.../recipes?title=sopa&sort=title&order=DESC

```
SELECT * FROM recipes
WHERE title LIKE "%sopa%"
ORDER BY title DESC
```



[...]




[...]



Ordenação

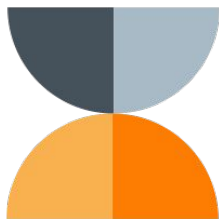
- Revisando as etapas de criação do endpoint, você percebe que basta fazer alguns ajustes e já pode partir para a implementação
 - Receber parâmetros por query string
 - Validar o **S** tipo **S** informado **S**
 - Consultar o banco usando **ORDER BY**
 - Verificar se algum valor foi retornado
 - Responder a requisição com o resultado da consulta

Vamos ver na prática! 



Pausa para relaxar 🧘

5 min



- Ordenação pode ser feita pelo frontend com o método sort, mas são bem mais diretas quando quando feitas com SQL

```
SELECT * FROM ... ORDER BY ... ASC/DESC
```



Paginação

Labenu_



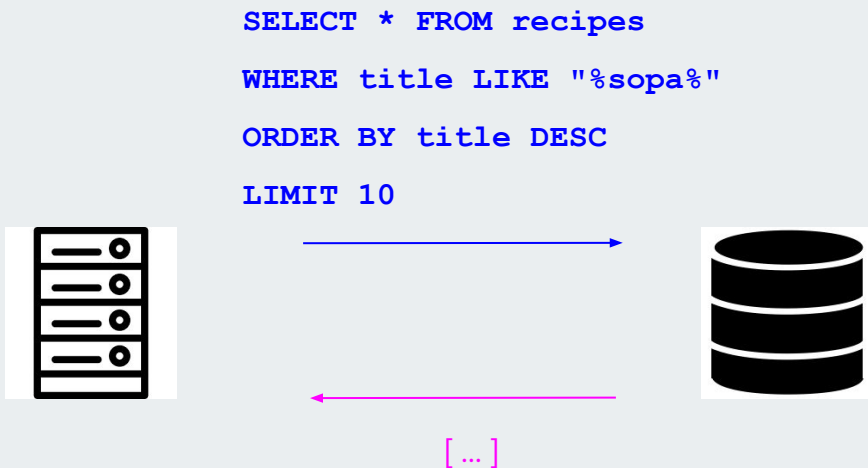
Paginação

- A refatoração do endpoint é um sucesso. Agora, resta apenas a questão do número de receitas retornado a cada consulta
- Para endereçar esse problema, você decide que as requisições retornarão ***páginas*** com 10 receitas cada, devendo incluir a numeração desejada (página 1 é o padrão)



Paginação 📖

- Você planeja implementar a regra de 10 receitas por página através da keyword LIMIT. No entanto, não sabe como obter a segunda página.

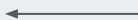


Paginação

- Pedindo ajuda aos colegas, eles o ensinam uma nova keyword: **OFFSET**.
- Com ela, você consegue desprezar os n primeiros resultados da consulta, possibilitando obter as páginas seguintes

```
SELECT * FROM recipes
WHERE title LIKE "%sopa%"
ORDER BY title DESC
LIMIT 10
OFFSET 20
```

Despreza os 20 primeiros itens
(retornando, assim, a terceira página)



Paginação

- Observando o padrão a seguir, você consegue deduzir a relação entre o número da página e o offset:
 - OFFSET 10 (10x1) retorna a página 2
 - OFFSET 20 (10x2) retorna a página 3
 - OFFSET 30 (10x3) retorna a página 4
 - OFFSET 40 (10x4) retorna a página 5

$$\text{offset} = \text{size} * (\text{page} - 1)$$



Paginação

- Voltando às suas anotações, você faz um último registro antes de partir para o código:
 - Receber parâmetros por query string
 - Validar o **S** tipo **S** informado **S**
 - Consultar o banco usando **ORDER BY** e **OFFSET**
 - Verificar se algum valor foi retornado
 - Responder a requisição com o resultado da consulta



Resumo

Labenu_



Resumo

- Um bom planejamento antes de começar a escrever o código pode economizar tempo, esforço e evitar frustrações
- **Filtros, ordenação e paginação** são atribuições típicas do backend, que é capaz de executá-las de maneira mais fácil e performática



Dúvidas? 🧐

Labenu_





Obrigado!