

Aula 48

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_



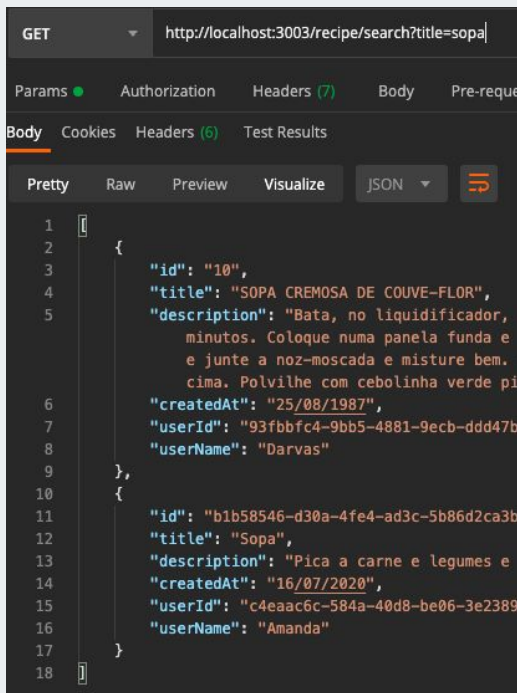
Filtros

- Pensando na escalabilidade da aplicação, o time de frontend solicita a adição de um endpoint que retorne as receitas **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?)



The screenshot shows a REST client interface with a GET request to `http://localhost:3003/recipe/search?title=sopa`. The response is displayed in the 'Body' tab, showing a JSON array of two recipe objects. The first object is for 'SOPA CREMOSA DE COUVE-FLOR' and the second is for 'Sopa'.

```
1 {
2   {
3     "id": "10",
4     "title": "SOPA CREMOSA DE COUVE-FLOR",
5     "description": "Bata, no liquidificador,
6       minutos. Coloque numa panela funda e
7       e junte a noz-moscada e misture bem.
8       cima. Polvilhe com cebolinha verde pi
9   },
10  {
11    "id": "b1b58546-d30a-4fe4-ad3c-5b86d2ca3b",
12    "title": "Sopa",
13    "description": "Pica a carne e legumes e
14    "createdAt": "16/07/2020",
15    "userId": "c4eaac6c-584a-40d8-be06-3e2389",
16    "userName": "Amanda"
17  }
18 }
```



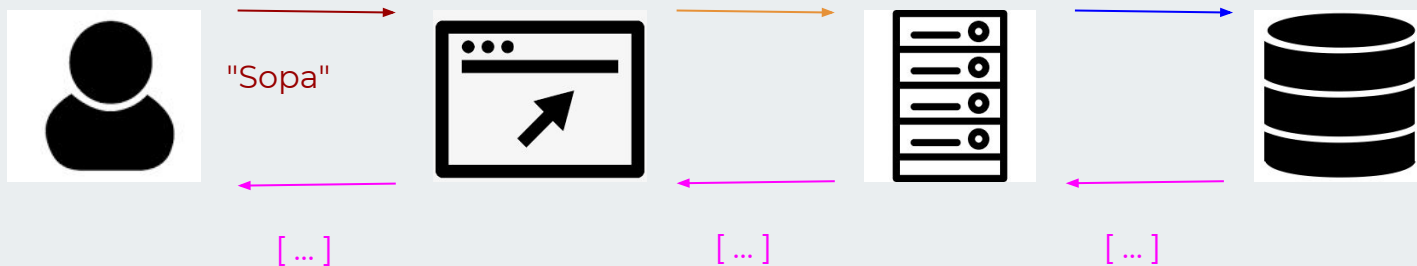
Filtros

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

`localhost:3003/recipe/search?title=sopa`

```
SELECT * FROM recipes
```

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





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

- Definir método e caminho do endpoint (GET, "/recipe/search")
- Receber o título da receita por *query string*
- Consultar o banco
- Responder a requisição com o resultado da consulta



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

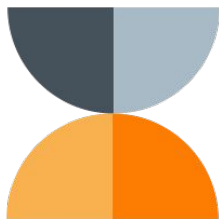
- Definir método e caminho do endpoint (GET, "/recipe/search")
- Receber o nome por *query string*
- **Validar o valor informado**
- Consultar o banco
- **Verificar se algum valor foi retornado**
- Responder a requisição com o resultado da consulta





4. PARTIR PARA A IMPLEMENTAÇÃO





- 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

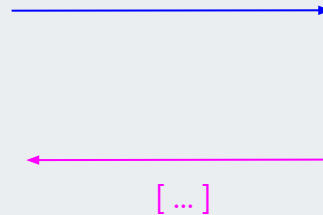
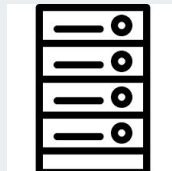
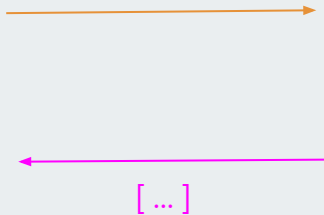
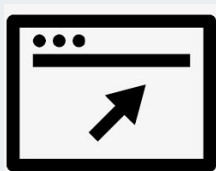


Ordenação 1 2 3

- Agora, a *query string* da requisição incluirá os parâmetros **orderBy**, para que os resultados possam ser ordenados por **título** ou **data de criação**, e **orderBy**, para que a ordenação possa ser **crescente** ou **decrescente**

.../search?title=sopa&orderBy=title&orderBy=DESC

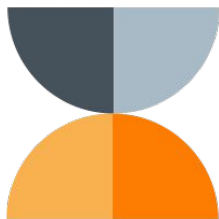
```
SELECT * FROM recipes  
WHERE title LIKE "%sopa%"  
ORDER BY title DESC
```



Ordenação

- Revisando as etapas de criação do endpoint, você percebe que basta fazer alguns ajustes e já pode partir para a implementação
 - Definir método e caminho do endpoint (GET, `"/recipe/search"`)
 - 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





- 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

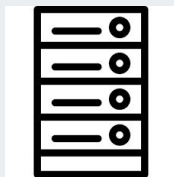
- A refatoração do endpoint é um sucesso. Agora, resta apenas a questão do número de receitas retornado a cada consulta
- Para endereçar esse problema, você decide que as requisições retornarão **páginas** com 5 receitas cada, devendo incluir a numeração desejada (página 1 é o padrão)



Paginação

- Você planeja implementar a regra de 5 usuários por página através da keyword LIMIT. No entanto, não sabe como obter a segunda página.

```
SELECT * FROM recipes  
WHERE title LIKE "%sopa%"  
ORDER BY title DESC  
LIMIT 5
```



[...]

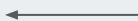


Paginação

- 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 5
OFFSET 10
```

Despreza os 10 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 5 (5x1) retorna a página 2
 - OFFSET 10 (5x2) retorna a página 3
 - OFFSET 15 (5x3) retorna a página 4
 - OFFSET 20 (5x4) retorna a página 5

$$\text{offset} = 5 * (\text{page} - 1)$$



Paginação

- Voltando às suas anotações, você faz um último registro antes de partir para o código:
 - Definir método e caminho do endpoint (GET, "/recipe/search")
 - Receber parâmetros por *query string*
 - Validar oS tipoS informadoS
 - 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





Obrigado!