Aula 17 - Mongo Avançado II

# Agregações

Consiste na realização de múltiplas operações, geralmente sobre múltiplos documentos.

Eles podem ser usados para:

* Agrupe documentos com base em critérios específicos.
* Realize alguma operação nos referidos documentos, a fim de obter um único resultado.
* Analisar as mudanças de informações ao longo do tempo.

**funcionando**

Os pipelines de agregação serão constituídos por um conjunto de etapas (estágios), onde cada etapa corresponderá a uma operação a ser realizada.

Podemos definir quantas etapas forem necessárias para atingir os resultados esperados.

Os documentos resultantes da etapa que se encerra são utilizados como “input” para a etapa seguinte, e assim sucessivamente até o final.

Um exemplo de pipeline de agregação pode ser:

1. Primeiro filtre os documentos que possuem um valor de x maior que 20
2. Em seguida, ordene-os do maior para o menor.
3. Então em um novo campo retorna o valor máximo
4. Então em um novo campo retorna o valor mínimo
5. Em seguida, em um novo campo, retorna a soma total de todos os documentos

**Principais estágios disponíveis em um pipeline de agregação**

$count : Conta o número de documentos disponíveis que estão no estágio atual.

$grupo: Permite agrupar os documentos disponíveis em novos grupos de acordo com um critério especificado. cada grupo tem um novo \_id, além dos valores acumulados.

$limit: Limita o número de documentos que sairão dessa etapa.

$lookup: Permite realizar um "left join" (combinação de campos), de uma coleção da mesma base de dados para os documentos da etapa atual.

$set / $addFields : Adiciona uma nova propriedade aos documentos que estão nessa etapa.

$skip: Retorna apenas os documentos que estão após o offset indicado.

$sort: Ordena os documentos na etapa atual.

$match: Retorna apenas os documentos que atendem a um critério de pesquisa, podemos colocar filtros comuns aqui

$merge: grava os resultados do pipeline em uma coleção. Deve ser o último estágio do pipeline para funcionar.

**paginação com mangusto**

**Paginação**

Quando trabalhamos pela primeira vez com nossos primeiros dados, é maravilhoso ver como nossas pesquisas podem retornar todos os dados de que precisamos.

Porém, essa “maravilha” começa a se tornar um problema quando a quantidade de dados que temos aumenta consideravelmente.

Lembremos que os dados que obtemos, no final temos que enviar pela internet, para que o cliente possa utilizá-los.

Quão lento será enviar 5.000 usuários de uma só vez? E 10 mil?

**Paginação como controle de resultado**

Aprender a pensar em páginas nos permitirá segmentar os resultados em pequenas informações, dando ao final uma referência de qual página estamos, qual é a página anterior e qual é a seguinte.

**Paginação usando mongoose-paginate-v2**

**mangusto-paginate-v2**

mongoose-paginate-v2 é um plugin para mongoose que nos permitirá realizar uma paginação eficiente para os modelos que especificamos.

Possui uma ótima otimização e adição de funcionalidades em comparação com sua v1

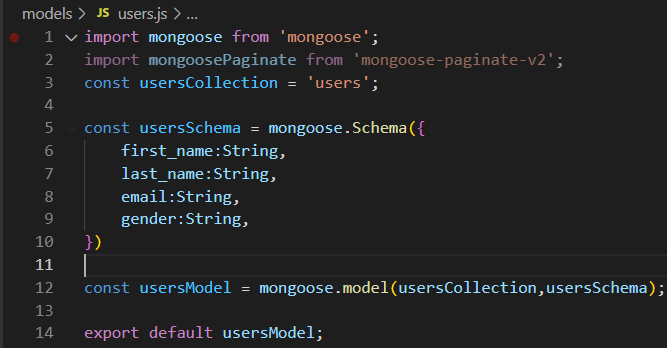
Para começar a usá-lo, basta instalá-lo com o npm:



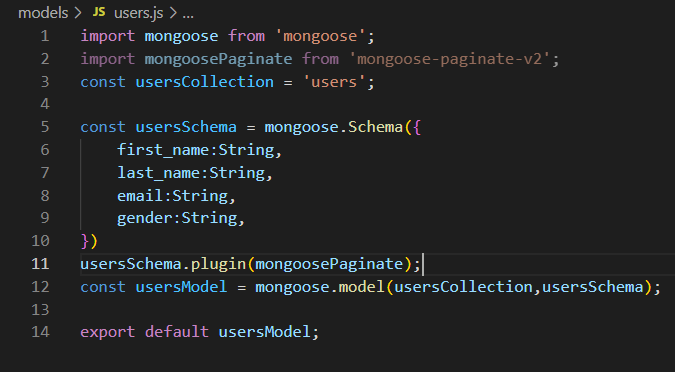
**Noções de mangusto-paginate-v2**

* docs: Os documentos retornados na página
* totalDocs: O total de documentos da consulta realizada.
* limit: Limite de resultados por página.
* page: Página atual onde estamos
* totalPages: Total de páginas que podem ser solicitadas na busca.
* hasNextPage: Indica se é possível avançar para uma próxima página.
* nextPage: Próxima página na busca
* hasPrevPage: Indica se é possível voltar para uma página anterior.
* prevPage: Página anterior na busca.
* pagingCounter: Número do documento em relação à página atual.

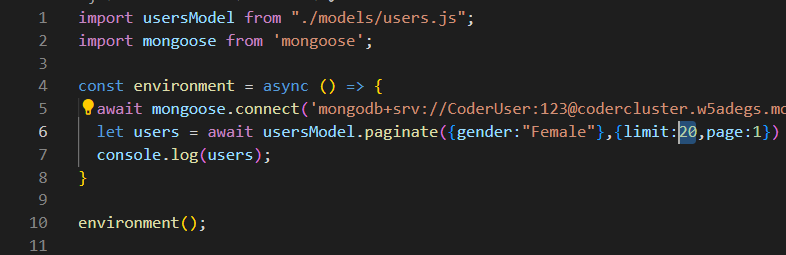
Para poder utilizá-lo, basta importar o módulo paginate no Schema onde iremos utilizá-lo.



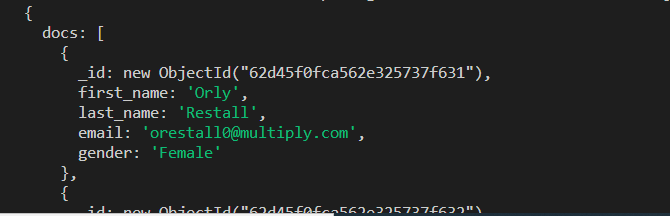
Portanto, antes de instanciar o modelo, adicionamos um plug-in de paginação ao nosso esquema

****

O modelo agora pode usar o método "paginar". O primeiro argumento é o filtro e o segundo são as opções.

****

Agora, no campo "docs" obteremos os resultados que solicitamos. Mas há mais!

****

Não apenas obtemos resultados, mas temos todas as informações sobre a paginação concluída e como podemos continuá-la.

