
API: Paginado

Gabriel Rodríguez Flores

December 2, 2021

- Filtro, orden y paginado

Contents

1	Teoría	3
1.1	Filtro	3
1.2	Ordenación	3
1.3	Paginado	3
1.3.1	Enlaces	3
2	Ejemplos	3
2.1	Ruta con Filtros	3
2.2	Ordenar elementos	3
2.3	Middleware de paginado	3
3	Ejercicios	4
4	Entregables	4
4.1	En clase	4
4.2	Tarea	5
4.3	Trabajo	5

1 Teoría

- Filtrado, ordenación y paginado
- Filtrado, ordenación y paginado en NodeJS

Nota: Toda esta lógica cambia cuando usamos BBDD

- Ejemplo de consultas SQL con paginado

1.1 Filtro

Se trata de definir por la query de la petición los parámetros y valores por los cuales queremos buscar los documentos.

Podemos hacer una búsqueda exacta o buscar un patrón o fracción del contenido.

- Ejemplo

1.2 Ordenación

Gestionar el sentido de la ordenación: `sort=parameter` o `sort=-parameter`

- Videotutorial

1.3 Paginado

1.3.1 Enlaces

2 Ejemplos

2.1 Ruta con Filtros

2.2 Ordenar elementos

2.3 Middleware de paginado

```
1 const paginationItems = (req, items, offset, limit) => {
2   const first = parseInt(offset, 10);
3   const step = parseInt(limit, 10);
4   const last = first + step;
```

```
5  const lastPage = Math.floor((items.total - 1) / step) * step;
6  const nextPage = Math.floor(last / step) * step;
7  const prevPage = Math.floor((first - step) / step) * step;
8
9  return {
10   links: {
11     base: `${req.headers.host}${req.path}`,
12     first: first !== 0 ? `?offset=0&limit=${step}` : undefined,
13     previous: first >= step ? `?offset=${prevPage}&limit=${step}` :
      undefined,
14     next: first < lastPage ? `?offset=${nextPage}&limit=${step}` :
      undefined,
15     last: first < lastPage ? `?offset=${lastPage}&limit=${step}` :
      undefined,
16   },
17   offset,
18   limit,
19   total: items.total,
20   pages: Math.ceil(items.total / step),
21   results: items.values,
22 };
23 };
24
25 module.exports = {
26   paginationItems,
27 };
```

3 Ejercicios

1. Crear servidor con un endpoint para recoger una lista de elementos (Ej. Notas, libros, usuarios, ...)
2. Sobre el servidor anterior, implementar:
 - Ordenación
 - Filtrado
 - Paginado

4 Entregables

4.1 En clase

Ejercicio 1

4.2 Tarea

Ejercicio 2

4.3 Trabajo

Incluir en el proyecto de notas la posibilidad de ordenar, filtrar y paginar las notas.

Ideas de implementación:

- Ordenación:
 - Por fecha de creación/edición
 - Por título
 - Por tamaño
- Filtro:
 - Contiene un texto en el título
 - Contiene un texto en el contenido
 - Igual a una categoría o grupo
 - Pertenece a un rango de fechas según creación/actualización
- Paginado:
 - Establecer un valor de elementos por página por defecto
 - Mostrar el total de elementos
 - Mostrar el total de páginas