

Manejo de parámetros de consulta

Elementos avanzados en tu API REST con Spring Boot

Hasta ahora

- Nuestras consultas han sido, o todos los productos, o uno por ID.
- Con la lección anterior, hemos limitado el número total de resultados, consiguiendo un subconjunto de todos página a página.
- Pero, ¿cómo obtener...
 - ...todos los productos de una categoría?
 - ...todos los productos que contengan una cadena de caracteres?
 - ...todos los productos cuyo precio sea menor a...?

Estructura de una URL: query

- Las URLs tienen, en su estructura, una parte denominada *query* (consulta).
- Viene a decirnos algo así que, *sobre un determinado recurso, queremos establecer unas condiciones de filtrado.*
- Es la parte final de la URL, y su comienzo viene determinado por un interrogante
- A partir de ahí, tenemos sucesivos parámetros clave=valor, separados por un ampersand, &.

Uso de parámetros en un controlador

- Con la anotación `@RequestParam` podemos recoger cualquier parámetro del controlador.
- Estos parámetros pueden ser obligatorios u opcionales.
- Podemos proporcionar un valor por defecto.

Uso de parámetros en un controlador

- **params:** este método del controlador solo llama si incluye ese (o esos) parámetro/s.
- **@RequestParam:** inyecta el valor en la variable de tipo String

```
@GetMapping(value = "/producto", params = "nombre")
    public ResponseEntity<?> buscarProductosPorNombre(
        @RequestParam("nombre") String txt,
        Pageable pageable, HttpServletRequest
request) { ... }
```

Servicio y Repositorio

- Si vamos a buscar por nombre, necesitamos un método de consulta en el repositorio

findByNombreContainsIgnoreCase(txt, pageable)

- En el servicio, creamos un *envoltorio*
- *findByNombre(String txt, Pageable pageable)*

Y ahora, unifiquemos todo el código

- Método en el controlador
- Método en el servicio
- Método en el repositorio
- Algún añadido en el manejo de errores

Algunas dificultades de este sistema

- Si queremos hacer la búsqueda por más de un parámetro, la consulta puede llegar a ser compleja
 - Condición compleja, uso de AND/OR
- La comprobación de qué parámetros hemos recibido y cuáles no nos obliga al uso de varios if/else.
- En la próxima lección veremos una solución más versátil, pero más compleja.

Reto

- Te animo a que puedas añadir algún término de búsqueda más (por ejemplo, precio menor que..., producto de una determinada categoría, ...).
- Recuerda que tendrás que
 - Crear la consulta en repositorio/servicio
 - Añadir el código en el controlador. Tendrás que modificar el método el método actual para que acepte más parámetros (te recomiendo refactorizar el nombre).