Ordenació de taules paginades

En moltes aplicacions web ens trobem amb pantalles que mostren llistats paginats. Per facilitar la feina a l'usuari podem afegir la possibilitat d'ordenar les dades fent servir com a referència la capçalera de la taula.

Així, cada cel·la de la capçalera es converteix en un enllaç que permet recarregar les dades ordenant-les segons la columna sobre la que s'actui.

El dipta-layout-starter facilita la implementació d'aquesta ordenació fent servir atributs específics a la plantilla HTML i l'interface Pageable a la part del controlador i l'accés a les dades.


```
<table class="table table-bordered table-sm table-sort">
    <thead>
    <tr>
        <th psd:table-sort="estat" psd:page-obj="${agrupacions}">Estat</th>
        <th psd:table-sort="assumpte">Assumpte agrupació</th>
        <th>Núm. expedients</th>
        <th psd:table-sort="audit.createdAt">Creació</th>
        <th psd:table-sort="audit.modifiedAt">Modificació</th>
        <th style="width: 1px" class="text-nowrap">
            Accions
        </th>
    </tr>
    </thead>
```

Ús de l'interface Pageable

La funcionalitat d'ordenació es basa en la utilització de l'interface Pageable que està definit en el paquet spring-data-web.

Aquest interface permet definir els criteris de paginació i d'ordenació d'una consulta i és comú tant per la part web (controladors) com per la part d'accés a dades (repositoris).

Fent ús de l'interface com a paràmetre dels mètodes del controlador ens evita haver de tractar de forma individual els paràmetres de paginació (número de pàgina, mida, etc) i els d'ordenació (camps d'ordenació, direcció, etc).

Els camps de la petició HTTP que fa servir l'interface Pageable són aquests:

- page: Número de la pàgina a mostrar, tenint en compte que el valor és "zero-indexed" i per tant la primera pàgina serà la 0
- size: El número d'items per pàgina. Per defecte és 20 tot i que es pot modificar per la configuració
- sort: Un o més paràmetres que es poden fer servir per ordenar el resultat. La direcció de l'ordenació s'afegeix al camp amb una coma (exemple: sort=name,asc)

Exemple

```
@LayoutController
@RequestMapping("/agrupacions")
public class AgrupacioController {

    @RequestMapping(value = "", method = RequestMethod.GET)
    public String index(Pageable pageable, Model model) {
        Page<Agrupacio> agrupacions = this.agrupacioService.obtenirAgrupacions(pageable);
        model.addAttribute("agrupacions", agrupacions);
        return "agrupacions/list.html";
    }
    ..
}
```

Aquest controlador acceptarà URLs com les següents:

- /agrupacions?page=1&size=15&sort=dataCreacio,desc
- /agrupacions?page=2
- /agrupacions?page=3&size=10&sort=name,asc

L'objecte Pageable "captura" totes aquests paràmetre generant un context de paginació i ordenació que serà enviat al servei per tal d'obtenir una pàgina de resultats. Aquesta pàgina es passarà a la vista en forma de "Page<?>" de tal manera que pugui ser tractada per l'atribut "table-sort" que s'encarrega de generar els enllaços de paginació

A la plantilla thymeleaf, a la capçalera de la taula de dades, podrem indicar quins camps es poden fer servir per l'ordenació mitjançant l'atribut "table-sort". A la taula també haurem d'indicar la classe "table-sort" per tal que els estils funcionin

```
<table class="table table-bordered table-sm table-sort">
    <thead>
    <tr>
        <th psd:table-sort="estat" psd:page-obj="${agrupacions}">Estat</th>
        <th psd:table-sort="assumpte">Assumpte agrupació</th>
        <th>Núm. expedients</th>
        <th psd:table-sort="audit.createdAt">Creació</th>
        <th psd:table-sort="audit.modifiedAt">Modificació</th>
        <th style="width: 1px" class="text-nowrap">
            Accions
        </th>
    </tr>
    </thead>
    <tbody>
    ...
    </tbody>
</table>
```

A l'exemple podem veure que:

- A la taula li hem afegit la classe "table-sort" per tal que després s'apliquin correctament els estils de la taula
- L'atribut "psd:table-sort" indica el camp d'ordenació de la columna. Per defecte s'agafa l'atribut del model que implementa l'interface Page<?> de Spring
- Si tenim més d'un objecte Page<?> per pantalla haurem de passar aqeust objecte mitjançant l'atribut "psd:page-obj"
- Les columnes que no tenen l'atribut psd:table-sort no seran ordenables

Valors per defecte i configuració

Si en una crida del controlador volem canviar el valor per defecte d'algun dels camps de paginació podem fer servir l'anotació @PageableDefault davant de l'objecte Pageable.

Per exemple:

```
public String index(@PageableDefault(sort = "nom", direction = Sort.Direction.ASC) Pageable pageable, Model model) {
```

en aquest cas, si no s'indiquen paràmetres d'ordenació, per defecte s'ordenarà pel camp "nom" de forma ascendent. Dins de l'anotació PageableDefault també podrien indicar altres camps, com la pàgina per defecte, o els items per pàgina

Si volem aplicar valors per defecte a tota l'aplicació i no només a una crida podem fer servir les propietats de configuració `spring.data.web.pageable`. Així, per definir que volem que tots els llistats siguin de 10 items (per defecte són de 20), aplicarem la següent configuració:

```
spring:
  data:
    web:
      pageable:
        default-page-size: 10
```

Estils

Els estils aplicats estan basats en els de Bootstrap4 de DataTables.net

Referències

- Baeldung.com: Spring Data Web Support
- Baeldung.com: Pagination and Sorting using Spring Data JPA

Tornar a l'índex