

Atelier 3

- Tri et pagination
- Validation
- Fichier template jumbotron-narrow
- Fichier css jumbotron-narrow.css
- Correction tri.

Tri et pagination

JpaRepository → PagingAndSortingRepository → CrudRepository

L'interface CrudRepository fournit les méthodes CRUD, PagingAndSortingRepository ajoute les méthodes de pagination et de tri, alors que l'interface JpaRepository définit en plus des méthodes spécifiques à la JPA.

1. Tri par rapport à une colonne :

```
Sort tri = new Sort(Direction.ASC, "nomChamp");
List<Intervenant> intervenants = intervenantRepository.findAll(tri);
```

2. Tri par rapport à plusieurs colonnes

```
Order ordre1 = new Order(Direction.ASC, "colonne1");
Order ordre2 = new Order(Direction.DESC, "colonne2");
Sort tri = new Sort(ordre1, ordre2);
List<Intervenant> intervenants = intervenantRepository.findAll(tri);
```

3. Pagination

```
int nombreLignes = 15;
int ligne = 0;
// Retourne nombreLignes à partir de la position page
Pageable page = new PageRequest(ligne, nombreLignes);
// On peut aussi trier
// Pageable page=new PageRequest(ligne,nombreLignes, sort);
Page<Intervenant> pageIntervenats = intervenantRepository.findAll(page);
List<Intervenant> intervenants = pageIntervenats.getContent();
```

La classe Page possède aussi les méthodes suivantes :

- getTotalElements()
- getTotalPages()
- hasNext()
- hasPrevious()

Remarque :

- les classes utilisées sont dans le package : org.springframework.data.domain

Exercice :

Modifier la vue « intervenants » qui affiche la liste des intervenants pour permettre le tri par rapport à l'email et au nom.

Validation

Java Bean Validation

- A partir de Spring 4, la spécification java beans validation JSR 349 est supportée

- Annotations définies : @Pattern, @Future, @Past, @Digital, @NotNull, @Null, @DecimalMax, @DecimalMin, @Size

Validation Hibernate

Annotations supportées : NotEmpty, Email, EAN, Length, Range, URL,NotBlank.

Pour la classe Intervenant, ajouter les annotations suivantes :

- @NotEmpty pour les champs email et nom
- @Email pour email
- @Size(min=2,max=30) pour nom

Dans le contrôleur IntervenantController ajouter l'annotation @Valid devant l'attribut intervenant de la méthode ajout :

```
@PostMapping("add")
public String enregistrer(@ModelAttribute @Valid Intervenant intervenant,
BindingResult result)
```

Intégration avec thymeleaf

L'attribut th:object="{intervenant}" doit être défini au niveau de l'élément form

Ajouter après la balise <form> :

```
<ul th:if="{#fields.hasErrors('*)}" class="alert alert-danger"
    role="alert">
  <li th:each="err: {#fields.errors('*)}" th:text="{err}" />
</ul>
```

Pour les champs nom et email ajouter un style qui sera appliqué dans le cas d'une erreur
th:errorclass="alert-danger"

Correction exercice tri

Template Jumbotron narrow

Index.html

```
/* Space out content a bit */
body {
  padding-top: 20px;
  padding-bottom: 20px;
}

/* Everything but the jumbotron gets side spacing for mobile first views */
.header,
.marketing,
.footer {
  padding-right: 15px;
  padding-left: 15px;
}

/* Custom page header */
.header {
  padding-bottom: 20px;
  border-bottom: 1px solid #e5e5e5;
}
/* Make the masthead heading the same height as the navigation */
.header h3 {
  margin-top: 0;
  margin-bottom: 0;
  line-height: 40px;
}

/* Custom page footer */
.footer {
  padding-top: 19px;
  color: #777;
  border-top: 1px solid #e5e5e5;
}

/* Customize container */
@media (min-width: 768px) {
  .container {
    max-width: 730px;
  }
}
.container-narrow > hr {
  margin: 30px 0;
}

/* Main marketing message and sign up button */
.jumbotron {
  text-align: center;
  border-bottom: 1px solid #e5e5e5;
}
.jumbotron .btn {
  padding: 14px 24px;
  font-size: 21px;
}

/* Supporting marketing content */
.marketing {
  margin: 40px 0;
```

```
}  
.marketing p + h4 {  
  margin-top: 28px;  
}  
  
/* Responsive: Portrait tablets and up */  
@media screen and (min-width: 768px) {  
  /* Remove the padding we set earlier */  
  .header,  
  .marketing,  
  .footer {  
    padding-right: 0;  
    padding-left: 0;  
  }  
  /* Space out the masthead */  
  .header {  
    margin-bottom: 30px;  
  }  
  /* Remove the bottom border on the jumbotron for visual effect */  
  .jumbotron {  
    border-bottom: 0;  
  }  
}
```

Fichier css : jumbotron-narrow.css

```
/* Space out content a bit */
body {
  padding-top: 20px;
  padding-bottom: 20px;
}

/* Everything but the jumbotron gets side spacing for mobile first views */
.header,
.marketing,
.footer {
  padding-right: 15px;
  padding-left: 15px;
}

/* Custom page header */
.header {
  padding-bottom: 20px;
  border-bottom: 1px solid #e5e5e5;
}
/* Make the masthead heading the same height as the navigation */
.header h3 {
  margin-top: 0;
  margin-bottom: 0;
  line-height: 40px;
}

/* Custom page footer */
.footer {
  padding-top: 19px;
  color: #777;
  border-top: 1px solid #e5e5e5;
}

/* Customize container */
@media (min-width: 768px) {
  .container {
    max-width: 730px;
  }
}
.container-narrow > hr {
  margin: 30px 0;
}

/* Main marketing message and sign up button */
.jumbotron {
  text-align: center;
  border-bottom: 1px solid #e5e5e5;
}
.jumbotron .btn {
  padding: 14px 24px;
  font-size: 21px;
}

/* Supporting marketing content */
.marketing {
  margin: 40px 0;
}
```

```
.marketing p + h4 {
  margin-top: 28px;
}

/* Responsive: Portrait tablets and up */
@media screen and (min-width: 768px) {
  /* Remove the padding we set earlier */
  .header,
  .marketing,
  .footer {
    padding-right: 0;
    padding-left: 0;
  }
  /* Space out the masthead */
  .header {
    margin-bottom: 30px;
  }
  /* Remove the bottom border on the jumbotron for visual effect */
  .jumbotron {
    border-bottom: 0;
  }
}
```

Correction exercice tri :

Modification de la méthode index dans le contrôleur :

```
@GetMapping
public String index(Model m) {
    m.addAttribute("intervenants", intervenantService.getIntervenants());
    return "intervenants/index";
}

@GetMapping("/{col}/{sens}")
public String indexAvecTri(Model m, @PathVariable String col, @PathVariable String
sens) {
    m.addAttribute("intervenants", intervenantService.getIntervenants(col, sens));
    return "intervenants/index";
}
```

La vue index.html

```
<!DOCTYPE html>
<html xmlns:th="http://thymeleaf.org"
      xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
      layout:decorator="layout/index">
<head>
<meta charset="UTF-8" />

</head>
<body>
    <div layout:fragment="titre">Intervenants</div>
    <div layout:fragment="content">

        <table class="table">
            <thead>
                <tr>
                    <th>Id</th>

                    <th><a href="/intervenants/nom/A">Nom</a></th>
                    <th><a href="/intervenants/email/A">Email</a></th>
                </tr>
            </thead>
            <tbody>
                <tr th:each="intervenant : ${intervenants}">
                    <td th:text="${intervenant.id}"></td>
                    <td th:text="${intervenant.nom}"></td>
                    <td th:text="${intervenant.email}"></td>
                    <!-- <td><a
th:href="@{details(id=${intervenant.id})}">Détails</a></td> -->
                    <!-- <td><a th:href="@{details/} +
${intervenant.id}">Détails</a></td> -->
                    <td><a
th:href="@{/intervenants/details/{id}(id=${intervenant.id})}"
class="glyphicon glyphicon-user"
/></td>

                    <td><a
th:href="@{/intervenants/del(id=${intervenant.id})}"
class="glyphicon glyphicon-scissors"
/></td>
```

```
        </tr>
      </tbody>
    </table>
  </div>
</body>
</html>
```