

## TANDABANY Adrien

### Compte rendu

#### Mission A

Pour afficher le détail du bombe, nous intervenons dans le controleur de Bombe.

Nous ajoutons la fonction show pour qu'elle puisse être exploitée dans le template show de Bombe.

```
@GetMapping("/admin/bombe/{id}")
fun show(@PathVariable id: Long, model: Model): String {
    val bombe = this.bombeDao.findById(id).orElseThrow()
    model.addAttribute("bombe", bombe)
    return "admin/bombe/show"
}
```

Le modèle va nous permettre d'afficher les données à la vue, et nous retourne le nom de la vue à afficher.

Puis dans le template show, nous affichons les différents attributs de la classe Bombe.

```
<p><strong>Bombe ID : </strong> <span th:text="${bombe.id}"></span></p>
<p><strong>Nom de la bombe : </strong> <span th:text="${bombe.nom}"></span></p>
<p><strong>Description de la bombe : </strong> <span th:text="${
{bombe.description}"></span></p>
<p><strong>Nombre de Dés : </strong> <span th:text="${bombe.nbrDes}"></span></p>
<p><strong>Dés Max : </strong> <span th:text="${bombe.maxDes}"></span></p>
```

On peut ajouter à la navbar dans le fichier mesFragments le chemin pour accéder à la Bombe depuis l'index.

```
<li><a class="dropdown-item" th:href="@{'/admin/bombe'}">Dashboard bombe</a></li>
```

Pour rappel, l'index permet d'afficher la page consulter. Voici la ligne suivante :

```
<a th:href="@{'/admin/bombe/'+${uneBombe.id}]" class="btn btn-primary">
    Consulter
</a>
```

#### Mission B

Ne sachant pas comment obtenir le nombre d'armures qui correspond à un type d'armure j'ai essayé de calculer nombre total d'armure pour me rapprocher du besoin demandé.

Dans la fonction index du controler TypeArmure, j'ai récupéré à nouveau tous les types d'armure comme ci-dessous dans une variable que je nommerai nombreTotalArmure :

```
val nombreTotalArmure = this.typeArmureDao.findAll()
```

Puis j'ai réalisé un model pour transmettre à la vue comme ci-dessous, en utilisant la méthode size :

```
model.addAttribute("nombreTotalArmure", nombreTotalArmure.size)
```

Et enfin sur la vue, j'affiche le nombre total d'armure comme indiqué ci-dessous :

```
<span th:text="Nombre total Armure +${nombreTotalArmure}"></span>
```

Notes : Je n'ai finalement pas réussi à faire cette expression de besoin.

## Mission C

Pour la dernière mission, je suis intervenu dans le contrôleur de personnage dans la fonction create, en modifiant le code ci-dessous :

```
@GetMapping("/joueur/personnage/create")
fun create(model: Model): String {
    val nouvellePersonnage = Personnage(null, "", 1, 1, 1, 1)
    val nouvelleArmure = Armure(null, "", "", "")
    val armure = armureDao.findAll()

    model.addAttribute("nouvellePersonnage", nouvellePersonnage)
    model.addAttribute("nouvelleArmure", nouvelleArmure)
    model.addAttribute("armure", armure)
    return "joueur/personnage/create"
}
```

J'ai ajouté une instantiation de Armure dans une variable nouvelleArmure puis son model et j'ai récupéré l'ensemble des armures avec le dao de l'armure dans une variable armure.

Enfin je modifie l'index pour faire une liste déroulante et afficher les armures :

```
<div class="mb-3">
    <label for="armure" class="form-label">Armure</label>
    <select class="form-select" id="armure" th:field="${nouvellePersonnage.armure.id}"
required>
        <option disabled selected value="">Choisir une armure</option>
        <option th:each="unType : ${armure}" th:value="${unType.id}" th:text="${
unType.nom}"></option>
    </select>
</div>
```

Notes : Pour cette mission je n'ai pas réussi à répondre à ce besoin par moi-même et j'ai dû faire appel à M. Moulin pour m'éclairer. J'ai essayé de m'appuyer sur les exemples que l'on peut retrouver dans le contrôleur et l'index Arme.