

ÉPREUVE E5 (session blanche 2024)

I – Épreuve

pour cette épreuve nous avons trois missions à réaliser dans un ordre précis (B-C-D)

A/ on souhaite que lors de la création d'un personnage (endpoint du formulaire : **/joueur/personnage/create**)

B/ faire en sorte de pouvoir afficher sur une nouvelle page le détail d'une **bombe**.

C/ dans le tableau des “**types d’armures**” faire apparaître dans une colonne sur le tableau le nombre “**d’armures**” qui correspond à chaque type d’armures.

II – Missions

a) mission B

pour cette mission il nous est demandé de créer une nouvelle page pour afficher en détail les informations de bombe. Pour y répondre, il a fallut créer les fichiers HTML dans le dossier bombes, principalement le fichier *show.html*.

```
<!DOCTYPE html>
<html lang="fr">
<head th:replace="mesFragments :: fragHead">
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
<div class="min-vh-100">
  <nav th:replace="mesFragments :: fragNav"></nav>
  <h1>Détails de la bombe</h1>
  <div>
    <p><strong>Nom : </strong> <span th:text="{bombe.nom}"></span></p>
    <p><strong>Description : </strong> <span th:text="{bombe.description}"></span></p>
    <p><strong> nbrDes: </strong> <span th:text="{bombe.nbrDes}"></span></p>
    <p><strong> maxDes: </strong> <span th:text="{bombe.maxDes}"></span></p>
  </div>
  <a th:href="@{'/admin/bombe/'+${bombe.id}+'/edit'}" class="btn btn-warning">
    Modifier
  </a>
  <a th:href="@{'/admin/bombe'}" class="btn btn-primary">
    Retour à la liste
  </a>
</div>
</div>
<footer th:replace="mesFragments :: fragFooter"></footer>
</body>
</html>
```

et de rajouter la méthode *show* dans le contrôleur de Bombe (Bombecontroleur) :

```
@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"
}
```

b) Mission C

pour faire apparaître le nombre d'armures qui correspond à chaque type d'armure il a fallut, d'abord, afficher la vue de armure dans son fichier index

```
<p>Nombres total d'armures : <span th:text="${nombreTotalArmures}"></span></p>
```

ajouter une méthode qui sert à compter le nombre d'éléments d'une liste par le contrôleur de armure

```
@GetMapping("/admin/armure")
fun index(model: Model): String {
    val armures = this.armureDao.findAll()
    model.addAttribute("armures", armures)
    model.addAttribute("nombreTotalArmures", armures.size)
    return "admin/armure/index"
}
```

c) Mission A

pour faire en sorte que le joueur puisse choisir une armure parmi la liste de toutes les amures, ajouter d'abord l'accès au DAO de armure dans le contrôleur de personnage :

```
/** DAO pour l'accès aux données des armures. */
val armureDao: ArmureDao,
```

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

    model.addAttribute("armures", valeurArmures)

    return "joueur/personnage/create"
}
```

```

@GetMapping("/joueur/personnage/{id}/edit")
fun edit(@PathVariable id: Long, model: Model): String {
    val personnage = this.personnageDao.findById(id).orElseThrow()
    val valeurArmure = armureDao.findAll()
    model.addAttribute("personnage", personnage)
    model.addAttribute("armure", valeurArmure)

    return "joueur/personnage/edit"
}

```

```

//ce que l'utilisateur récupère
personnageModifier.armureEquipee = personnage.armureEquipee

```

dans le *create.html* ajouter :

```

<!-- Ajoutez d'autres champs du formulaire si nécessaire -->
<div class="mb-3">
    <label for="armures" class="form-label">Armure</label>
    <select class="form-select" id="armures" th:field="{nouvellePersonnage.armureEquipee.id}">
        <option th:each="uneArmure : ${armures}" th:value="{uneArmure.id}" th:text="{uneArmure.nom}"></option>
    </select>

```