

Ecole Supérieure Privée Technologies & Ingénierie

Type d'épreuve : ☐ Devoir ☒ Examen
Enseignant : Arfaoui Ahlem
Matière : Framework Spring Boot
Année Universitaire : 2023-2024 Semestre : 1
Classe : CII-2-GLSI
Documents : ☐ Autorisés ☐ Non autorisés ☒
Date : 05/01/2024 Durée : 1h :30
Nombre de pages : 7 pages

Exercice 01 :

A. QCM :

1. Quelle est la bonne syntaxe qui permet d'afficher un lien dans une page Thymeleaf :
 - a. `@{link}`
 - b. `#{link}`
 - c. `*{link}`
 - d. `${link}`
2. Quelle technologie est utilisée dans Spring Boot pour la persistance des données ?
 - a. JDBC
 - b. JAXWS
 - c. JPA
 - d. JMS
3. Quels types d'injection de dépendance que Spring prend-il en charge ?
 - a. Basé sur le constructeur et les setters.
 - b. Basé sur le constructeur, les setters, et les getters.
 - c. Basé sur les setters, les getters, et les propriétés.
 - d. Basé sur le constructeur, les setters, et les propriétés.
4. Le langage d'expressions (ou Expression Language) fait partie de Core Container dans le Framework Spring ?
 - a. Vrai
 - b. Faux
5. Lequel des énoncés suivants est correct concernant le Framework Spring ?
 - a. Le Framework Spring est une solution lourde (heavy-weight solution).
 - b. Le Framework Spring est une solution légère (light-weight solution).

- c. Les deux a et b sont vrais.
- d. Aucune de ces réponses n'est vraie.

6. Quel est le moteur de Template HTML par défaut dans Spring Boot ?

- a. JSP
- b. HTML
- c. Thymeleaf
- d. Toutes les réponses sont vraies

B. Question/Réponse :

1. Quelle est la différence entre **@Controller** et **@RestController** dans Spring Boot ?

.....

.....

.....

.....

2. Quelle est la différence entre une application **Spring** et une application **Spring Boot** ?

.....

.....

.....

.....

3. Quand utiliser **@Autowired** dans une classe Spring Boot ?

.....

.....

.....

4. Quels sont les deux systèmes d'authentification et expliquer la différence entre ces derniers ?

.....

.....

.....

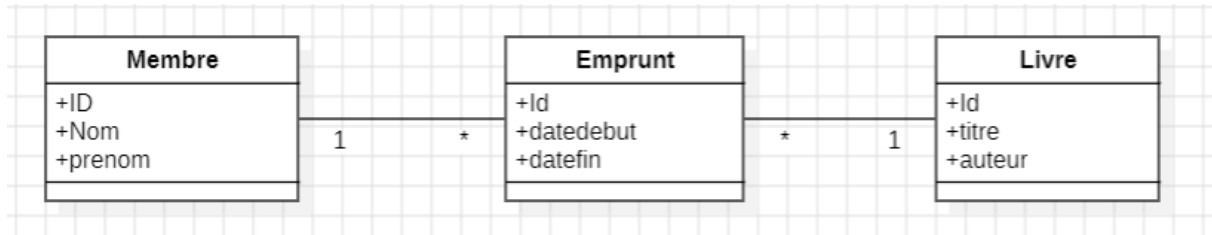


Exercice 02 :

Remarque : N'oubliez pas d'ajouter les annotations.

Notre objectif consiste à développer une application dédiée à la gestion efficace d'une bibliothèque, offrant aux bibliothécaires des fonctionnalités. Pour ce faire, nous envisageons d'utiliser le Framework Spring Boot pour construire une solution robuste et évolutive.

Soit le diagramme de classes suivant :



A. Les entités :

A.1. Terminer les entités suivantes. N'oubliez pas d'ajouter les annotations.

[illegible]

B. L'ajout et l'affichage de la liste des Emprunts :

Soit **EmpruntRepository**, le répertoire qui contient les fonctionnalités de gestion des Emprunts.

Soit **EmpruntService**, le service de gestion des Emprunts.

Soit **EmpruntController**, le contrôleur responsable des actions de gestion des Emprunts.

B.1. Ajouter la classe **EmpruntService** :

```
//Import
```

```
.....
```

```
Public class EmpruntService {
```

```
.....
```

```
.....
```

```
//Ajouter l'Emprunts
```

```
.....
```

```
.....
```

```
.....
```

```
//La liste de toutes les Emprunts
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
}
```

B.2. Terminer la classe **EmpruntController**.

```
//Import
```

```
.....
```

```
.....
```

```
Public class EmpruntController {
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
//ajouter Emprunt et redireger vers la liste des Emprunts
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

}

 id |

C. Update et Delete d'un Livre :

Soit **LivreRestController**, le contrôleur (Est un Contrôleur Rest) responsable des actions de gestion des Livres.

Soit **LivreRepository**, le répertoire qui contient les fonctionnalités de gestion des livres.

Soit **LivreService**, le service de gestion des Livres.

C.1. Terminer la classe **LivreService**

```
//Import....
```

```
Public class LivreService {
```

//Mise à jour d'un livre

```
//supprimer un livre
```

}

C.2. Terminer la classe **LivreRestController**.

```
//Import.....
```

```
.....
```

```
Public class LivreRestController {
```

```
.....
```

```
//Mise à jour d'un livre
```

```
.....
```

```
//supprimer un livre
```

```
.....
```

```
}
```