
Matière	: Spring Boot		
Enseignant	: Mohamed Anouer DAHDEH		
A.U	: 2021-2022	Semestre	: 2 (Session principale)
Classe	: GLSI 2		
Documents	: Non autorisés		
Date	: 30/05/2022	Durée	: 1H
Nombre de pages	: 3 pages		

Partie 1 : QCM (5pts)

Choisir la bonne réponse :

Q1 : Comment identifier un fragment dans Thymeleaf ?

- a. Sélectionner toutes les étiquettes (tag) avec l'attribut `th:fragment="name-fragment"`
- b. Sélectionner l'étiquette par ID
- c. Sélectionner toutes les étiquettes par Css-class
- d. Toutes les réponses sont correctes

Q2 : Quel composant fait partie de la couche d'accès aux données dans le Framework Spring ?

- a. Enum
- b. Repository
- c. Beans
- d. Controller

Q3 : Pour une entité **Car (id, model, year)**, laquelle de ces méthodes est automatiquement synthétisée dans **JpaRepository** ?

- a. `findCarByModel()`
- b. `findByModelAndYear()`
- c. `findByModel()`
- d. `findModel()`

Q4 : Quelle annotation est utilisée pour initialiser l'argument de la méthode associée à l'URI <http://localhost/students/{id}> par la valeur de **id**

- a. `@Put`
- b. `@RequestBody`
- c. `@RequestParam`
- d. `@PathVariable`

Q5 : Lors de la génération d'une vue par le Thymeleaf Engine, laquelle des réponses est correcte à propos l'attribut **th:classappend** :

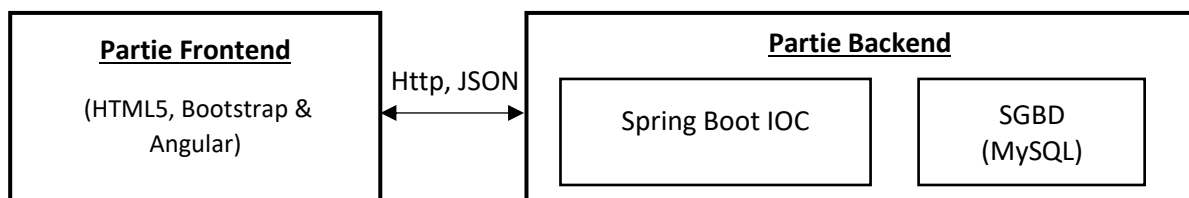
- a. Permet de créer l'attribut **class**
- b. Permet de remplacer l'attribut **class** disponible
- c. Permet d'ajouter des valeurs pour l'attribut de **class**
- d. Toutes les réponses sont correctes

Partie 2 : (15pts)

On souhaite mettre en place une application web de gestion des inscriptions pour un club informatique au sein de la faculté. La méthodologie de travail adoptée est Scrum. La phase préparatoire du projet a conduit au Backlog du produit suivant :

Sprint	Id	User Story		Estimation
		En tant que ...	Je veux ...	
1	1	Adhèrent	M'inscrire au club	3
	2	Responsable du club	Afficher la liste détaillée des inscriptions	3
	3	Responsable du club	Supprimer une inscription	1
2	4	Responsable du club	M'authentifier	8

L'architecture logicielle adoptée est présentée par la figure suivante :



Lors de son inscription, un adhérent doit remplir un formulaire contenant les informations obligatoires suivantes : numéro de carte étudiant (unique), nom, prénom, email (unique) et classe. Ces données sont enregistrées dans la table « inscriptions ».

```

inscription_db.inscriptions
id : bigint(20)
classe : varchar(10)
email : varchar(50)
nom : varchar(30)
num_carte : varchar(10)
prenom : varchar(30)
  
```

Cet extrait montre la structure de la table 'inscriptions' dans une base de données. Les champs sont : id (bigint(20)), classe (varchar(10)), email (varchar(50)), nom (varchar(30)), num_carte (varchar(10)) et prenom (varchar(30)).

Travail à faire :

En tant que membre de l'équipe Scrum et développeur Backend, vous allez créer une API REST afin d'accomplir votre fiche TODO contenant les 3 user stories **du sprint 1** :

- 1- Quels sont les straters nécessaires à ajouter au fichier pom.xml ?
- 2- Proposer à l'aide d'un schéma une architecture multicouche pour l'API REST.
- 3- Lister dans chaque couche, les packages ainsi que les classes à implémenter.

- 4- Compléter le tableau suivant par les requêtes http (URI), correspondantes pour chaque User Story :

User story	Verbe HTTP	URI
1		http://localhost :8080/....
2		http://localhost :8080/....
3		http://localhost :8080/....

- 5- Donner le code des différentes classes nécessaires pour implémenter le 2^{ème} user story

Consignes :

- Si la liste est vide l'API REST doit retourner le message « Aucune inscription trouvée » sinon une liste des inscriptions est envoyée à la partie frontend
- Pour ajouter une contrainte d'unicité, il faut utiliser l'attribut unique dans l'annotation @Column comme suit : @Column (unique=true)

Bon Travail