

Ecole Supérieure Privée Technologies & Ingénierie

Type d'épreuve	: <input type="checkbox"/> Devoir <input checked="" type="checkbox"/> Examen	
Enseignant	: Mohamed Anouar DAHDEH	
Matière	: JEE	
Année Universitaire	: 2019-2020	Semestre : 1
Classe	: CII-2 GLSI-C	
Documents	: <input type="checkbox"/> Autorisés	<input checked="" type="checkbox"/> Non autorisés
Date	: 15/02/2020	Durée : 1h30mn
Nombre de pages	: 3 pages	
Barème	: (5pts +15pts)	

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. Entitie

Q3 : Pour une classe d'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 mapper **value** à l'argument de la méthode dans l'URI :

`http://localhost/students/{value}`

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

Q5 : Dans Thymeleaf, lors de la génération HTML par le Thymeleaf Engine, laquelle des réponses est correcte à propos l'attribut **th:classappend** :

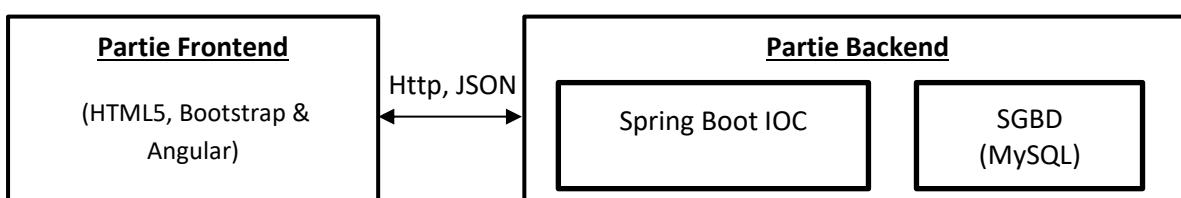
- Permet de créer l'attribut **class**
- Permet de remplacer l'attribut **class** disponible
- Permet d'ajouter des valeurs pour l'attribut de **class**
- Toutes les réponses sont correctes

Partie 2 : (15pts)

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

Sprint	Id	User Story		Estimation
		En tant que ...	Je veux ...	
1	1	Adhérent	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)

Travail à faire :

En tant que membre de l'équipe Scrum et développeur Backend, vous allez créer un 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 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

Remarques :

- 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 à une colonne de la table « inscriptions » il faut utiliser l'annotation @Column (unique=true)

- 6- Lister les trois techniques utilisées pour sécuriser une API REST.

Bon Travail