

Semestre : 1 ☒ 2 ☐

Session : Principale ☒ Rattrapage ☐

Module : Services Web

Enseignant(s) : UP_WEB

Classe(s) : 4A

Documents autorisés : OUI ☐ NON ☒ Nombre de pages : 9

Calculatrice autorisée : OUI ☐ NON ☒ Internet autorisée : OUI ☐ NON ☒

Date : 19/01/2022 Heure : 09h00

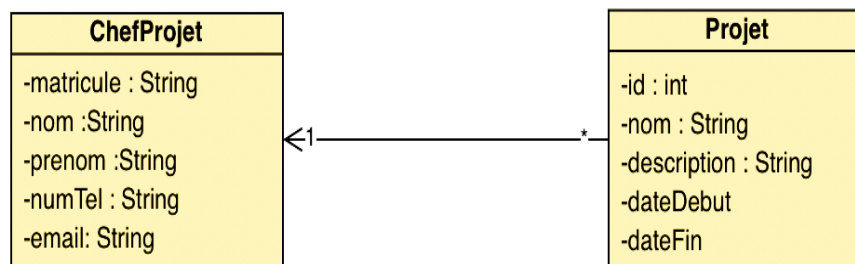
Durée : 1h30

Exercice 1 (8pts)

Etude de cas : Système de suivi des projets

Dans le but d'aider les entreprises de développement à suivre et gérer les projets, on se propose de mettre en place un service Web leur permettant d'ajouter, consulter les projets, se renseigner sur leurs date de début et fin ainsi que d'avoir une supervision continue par un chef de projet.

Un projet possède les informations suivantes : un identifiant, un nom, une description, une date de début, une date de fin et un Chef de projet. Ce dernier est caractérisé par un matricule (son identifiant), un nom et prénom, un numéro de téléphone et un email.



Les ressources REST qui constituent le service web sont présentées ci-dessous.

Soit les descriptions suivantes de l'API REST :

- **Ajout d'un projet**

Requête	POST	/projets
	Header	Content-Type: application/json
	Body	{ " id": "1", " nom": "Application web", " description": "développement d'une application web avec Spring boot et Angular", " dateDebut": "23/10/2021", " dateFin": "23/10/2022", " chefProjet": { " matricule": "matricule123", } }
Réponse	Header	Content-type : text/plain
	Status	201 CREATED / 404 NOT FOUND
	Body	Projet a été ajouté avec succès

- **Récupération de la liste de tous les projets.**

Requête	GET	/projets
Réponse	Header	Content-Type: application/json
	Status	200 OK / 404 NOT FOUND
	Body	[{ " id": "1", " nom": "Application web", " description": "développement d'une application web avec Spring boot and Angular", " dateDebut": "23/10/2021", " dateFin": "23/10/2022", }

		<pre> "chefProjet": { "matricule": "matricule123", "nom": "Mohamed", "prenom": "Salah", "email": "mohamedsalah@gmail.com", "numtel": "12345678" }, { "id": "2", "nom": "Application mobile", "description": "développement d'une application mobile avec Spring boot et React Native", "dateDebut": "01/10/2021", "dateFin": "01/10/2022", "chefProjet": { "matricule": "matricule456", "nom": "Karim", "prenom": "Ben Mohamed", "email": "Karimbenmohamed@gmail.com", "numtel": "87654321" } }] </pre>
--	--	--

- Récupération de la liste des projets pour un chef de projet spécifique.

Requête	GET	/projets?matricule=matricule456
Réponse	Header	Content-Type: application/json
	Status	200 OK / 404 NOT FOUND
	Body	[{

		<pre> "id": "2", "nom": "Application mobile", "description": "développement d'une application mobile avec Spring boot et React Native", "dateDebut": "01/10/2021", "dateFin": "01/10/2022", "chefProjet": { "matricule": "matricule456", "nom": "Karim", "prenom": "Ben Mohamed", "email": "Karimbenmohamed@gmail.com", "numtel": "87654321" } }] </pre>
--	--	---

- Modification d'un projet ayant un identifiant spécifique.

Requête	PUT	/projets/2
	Header	Content-Type: application/json
	Body	<pre> { "id": "2", "nom": "Application Cross Plateforme", "description": "développement d'une application mobile avec Spring boot et React Native", "dateDebut": "01/10/2021", "dateFin": "01/10/2022", "chefProjet": { "matricule": "matricule123", } } </pre>
Réponse	Status	200 OK
	Body	true / false

Travail demandé :

1) Développez le service web RESTful décrit ci-dessus en complétant les parties manquantes des pseudos codes suivants dans le tableau ci-dessous : (4pts)

a) La classe **RestActivator** :

```
import javax.ws.rs.core.Application;
@..... [1]
public class RestActivator extends Application {

}
```

b) Quel est le rôle de la classe ci-dessus ? **RestActivator**.

c) La classe **ProjetRessource** où nous allons implémenter les sous-ressources de la gestion des projets.

```
@.....[2]
public class ProjetRessource {
    public static List<Projet> projets=new ArrayList<Projet>();

    @.....[3]
    @.....[4]
    @.....[5]
    public .....[6] ajouterProjet(Projet p) {
        if(projets.add(p)) {
            return Response.status(Status..... [7] );
        }
        return Response.status(Status..... [8] );
    }

    @.....[9]
    @.....[10]
    @.....[11]
    public .....[12] modifierProjet(..... [13] ,.....[14] ) {
        int index= this.getIndexById(id);
        if (index!=-1) {
            projets.set(index, p);
            return .....[15];
        }
        return .....[16];}
}
```

```

    public int getIndexById(int id) {
        for(Projet p: projets) {
            if (p.getId()==id)
                return projets.indexOf(p);
        }
        return -1;
    }
}

```

2) Soit la continuité de la classe **ProjetRessource** ci-dessous :

```

//.....
public static List<Projet> projets=new ArrayList<Projet>();
//.....
@GET
@Produces(MediaType.APPLICATION_JSON)
public Response RecupererProjetListe() {
    if(projets.size()!=0) {
        return Response.status(Status.OK).entity(projets).build();
    }
    else return Response.status(Status.NOT_FOUND).build();
}

@GET
@Produces(MediaType.APPLICATION_JSON)
public Response RecupererProjetListebyChef(@QueryParam(value="matricule") String
matricule) {
    List<Projet> liste=new ArrayList<Projet>();
    for(Projet p:projets){
        if(p.getChefProjet().getMatricule().equals(matricule))
            liste.add(p);
    }
    if(liste.size()!=0) {
        return Response.status(Status.OK).entity(liste).build();
    }
    else
        return Response.status(Status.NOT_FOUND).build();
}
//.....

```

Au cours des tests de l'API , nous avons rencontré un problème. **Le serveur nous renvoie une erreur de HTTP Status 500 – Internal Server Error.**

a) Expliquer l'origine de cette erreur. (1pt)

.....

b) Développez une solution pour la corriger en respectant la description du web service.
(2pts)

3) Supposant qu'on va ajouter la fonctionnalité de la suppression d'un projet bien déterminé avec la méthode **supprimerProjet** décrit ci-dessous : (1pt)

```
@DELETE
public Boolean supprimerProjet(@QueryParam(value="id") int id) {
    int index= getIndexById(id);
    if (index!=-1) {
        projets.remove(index);
    }
    return True;
} else
return False;
}
```

Donner la description de la requête HTTP permettant de consommer le service ainsi que la réponse HTTP retournée avec un exemple concret.

Requête	
Réponse	

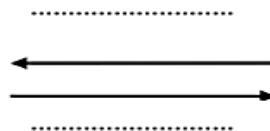
Exercice 2 (1pt)

Complétez les différentes transformations de l'API JAXB avec les mots de la liste :

serialize - wsimport - schemagen - deserialize - javac - marshal – xjc - unmarshal- xsdgen

Document XML

```
<individual>
<id>1</id>
<firstname>
John
</firstname>
<homeAddress>
...
</homeAddress>
</individual>
```



Instance d'objet



Exercice 3 (11pts)

Sélectionnez la réponse correcte et répondez aux questions (6, 7 et 8) :

- 1) On dit que XML est un méta langage parce qu'il :
- a. permet la séparation du contenu de la forme
 - b. permet au développeur de créer ses propres balises
 - c. permet de définir d'autres langages dérivés
 - d. toutes les réponses sont correctes
- 2) Un document XML doit avoir :
- a. Des attributs
 - b. Un élément racine
 - c. Un prologue
 - d. Toutes les réponses sont correctes
- 3) En XML, lequel des noms de balises suivantes est syntaxiquement correct ?
- a. <2emeBatiment>
 - b. <_le2emeBatiment>
 - c. <le2eme Batiment>
 - d. Aucune réponse
- 4) Les consommateurs et les fournisseurs de services web échangent des flux de données sous format ?
- a. XML
 - b. Json
 - c. SOAP
 - d. Plusieurs formats de données
- 5) En raison de leur petite taille, les JWT peuvent être envoyés via :
- a. Une URL,
 - b. Un paramètre POST
 - c. Dans un en-tête http
 - d. Les 3 réponses sont correctes

6) Citez trois caractéristiques d'un service web

.....

7) Donnez deux avantages d'utiliser l'API JAX-B

.....

8) Rajoutez les annotations JAX-B nécessaire afin d'avoir le document XML correspondant.

Affectation.java

```
public class Affectation {  
    protected String supervisor;  
    protected String class;  
    protected Date date;  
    protected String Subject;  
    .....  
}
```

Affectation.xml

```
<affectation date="19/01/2022">  
  <surveillant>Wael Najjar</surveillant>  
  <classe>4A12</classe>  
</affectation>
```

Bon Travail