

TP N°1 First Micro-service With_SpringBoot

Soit une application pour la gestion d'une plateforme Cloud, tel que :

- Un **Utilisateur** est défini par un Id, un nom, un email et il peut créer plusieurs **Machines Virtuelles**.
 - Une **Machine Virtuelle** est définie par un Id, une **configuration** (CPU, RAM, DISCK-SIZE) et elle est déployée dans deux **Serveurs (primary, backup)**.
 - Un **Serveur** a un id, une **configuration** (CPU, RAM, DISCK-SIZE) et le nom du datacenter auquel se trouve. Un **Serveur** peut déployer plusieurs **Machines Virtuelles**.
 - Les configurations doivent être décrites dans des **Embeddable Classes**.
1. Créer un projet SpringBoot en utilisant spring initializr et choisir les dépendances suivantes :
 - a. Spring Web, Spring REST Repositories, Spring Data JPA, Mysql Driver, Lombok.
 2. Créer les entités correspondantes en prenant en considération les contraintes suivantes :
 - Tous les noms sont obligatoires et d'une longueur de 20 caractères
 - L'email : unique
 - Pour les utilisateurs, les machines virtuelles sont chargés immédiatement (EAGER)
 - Les configurations sont chargées à la demande (Lazy)
 3. Réaliser la couche DAO et Insérer des données dans les trois entités : département, projet et employé.
 4. Etendre la couche Repository en ajoutant les méthodes suivantes :
 - a. **Query Derivation** :
 - i. Créer une méthode qui retourne les utilisateurs dont l'email appartient à un domaine donné, ex : « esi-sba.dz »
 - b. **Annotated Query** :
 - i. Créer une méthode qui retourne l'IdUser et le nombre de ses machines virtuelles
 5. Créez une **projection** qui retourne pour chaque serveur son nom, son CPU, sa RAM et le nom de son datacenter.
 6. Testez les différents Paths avec les différentes méthode http (GET, POST, PUT, PATCH, DELETE) générées par **@RepositoryRestResource**.

7. Créez une API REST personnalisée en utilisant l'annotation **@RestController**. Cet API expose les Paths suivants :

- a. GET <http://localhost:8081/rsu/serveur/idServeur/vms> : retourne les VMs qui sont déployées (primary & backup) dans le serveur idServeur
- b. POST <http://localhost:8081/rsu/vm> : Ajoute une VM.

```
{  
  "idUtilisateur":"1",  
  "idServeur":"2",  
  "idServeurBackup" : "1",  
  "configuration":{  
    "cpu":"2", "ram":"16", "disk":"100"  
  }  
}
```

- c. PUT <http://localhost:8081/rsu/user/{id}> : modifie un utilisateur en indiquant son Id (le tout est défini dans le *RequestBody*)
- d. PATCH <http://localhost:8081/rsu/user/{id}> : modifie un utilisateur en indiquant son Id (le tout est défini dans le *RequestBody*)
- e. DELETE <http://localhost:8081/rsu/serveur/{idserveur}/{idvm}> : supprime la VMs IdVM du serveur IdServeur.