

Atelier REST n°3

JAX_RS CRUD

Objectifs

Notre objectif est de créer un service web RESTful qui fournira un exemple d'opérations CRUD (Créer, lire, mettre à jour et supprimer).

Développement du service web RESTful

Nous visons à créer un service web de gestion des employés dans une entreprise.

| Employe |
|---|
| -cin:String -nom:String -prenom:String |
| ajouterEmploye(Employe employe) modifierEmploye(Employe employe) chercherEmploye(String cin) afficherListeEmployes() supprimerEmploye(String cin) |

1. Créez un projet **maven** de packaging **war**.
2. Créez l'entité "**Employe**" comme suit:

```

@XmlRootElement
public class Employe {
    private String cin;
    private String nom;
    private String prenom;

    public String getCin() {
        return cin;
    }
    @XmlAttribute(name="id",required=true)
    public void setCin(String cin) {
        this.cin = cin;
    }
    public String getNom() {
        return nom;
    }
    @XmlElement(name="LastName")
    public void setNom(String nom) {
        this.nom = nom;
    }
    public String getPrenom() {
        return prenom;
    }
    @XmlElement(name="FirstName")
    public void setPrenom(String prenom) {
        this.prenom = prenom;
    }
    public Employe() {
        super();
        // TODO Auto-generated constructor stub
    }
}

```

- au niveau de l'entité "**Employee**", générez les deux méthodes equals et hashCode.

*equals () est utilisée dans la plupart des collections pour déterminer si une collection contient un élément donné.

* hashCode () est utilisée pour déterminer où stocker et où chercher l'objet en interne.

3. Créez les ressources REST qui constituent le service web selon les descriptions suivantes:

a) addEmployee

| |
|--|
| Requête POST /employees Content-Type: application/xml <pre><employee id="123"> <FirstName>foulen</FirstName> <LastName>ben foulen</LastName> </employee></pre> |
| Réponse add successful Status 200OK |

- testez la méthode avec un outil de test des requêtes HTTP(Advanced REST client, poster, postman...)

b) updateEmployee

| |
|---|
| Requête PUT /employees Content-Type: application/xml <pre><employee id="123"> <FirstName>ahmed</FirstName> <LastName>ben salah</LastName> </employee></pre> |
| Réponse update successful Status 200OK |

c) searchEmployee

| |
|--|
| Requête GET /employees/123 |
| Réponse <pre><employee id="123"> <FirstName>foulen</FirstName> <LastName>ben foulen</LastName> </employee></pre> Status 200OK |

d) displayEmployeesList

| |
|--|
| Requête GET /employees |
| Réponse <collection> <employee id="123"> <LastName>ben foulén</LastName> <FirstName>foulén</FirstName> </employee> <employee id="456"> <LastName>ben foulén2</LastName> <FirstName>foulén2</FirstName> </employee> </collection> Status 200OK |

e) deleteEmployee

| |
|---|
| Requête DELETE /employees/123 |
| Réponse true Status 200OK |

4. Renvoyez la requête de suppression:

| |
|--|
| Requête DELETE /employees/123 |
| Réponse false Status 200OK |

- a. Que remarquez-vous?
- b. Comment résoudre ce problème?