

Atelier REST n°1**Mise en place d'un service web RESTFul****Objectifs**

On se propose de développer le service web de logement présenté dans la situation problème globale (prosit initial).

Deux alternatives de logement sont proposées aux étudiants faisant leur inscription administrative à travers le portail. La première option consiste à faire une réservation dans la résidence universitaire de l'école. La deuxième possibilité consiste à faire une réservation auprès des agences immobilières spécialisées pour les étudiants et qui proposent des logements conventionnés.

L'objectif de cet atelier de développer le service web RESTFul de logement conventionné.

Contexte

Le projet qui vous a été fourni contient le code d'implémentation des entités ainsi que le traitement métier.

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

Logement
- reference: int
- adresse: String
- delegation: String
- gouvernorat: String
- type: String
- description: String
- prix: float

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

```
import javax.xml.bind.annotation.XmlRootElement;
@XmlRootElement
public class Logement {
    private int reference;
    private String adresse;
    private String delegation;
    private String gouvernorat;
    private String type;
    private String description;
    private float prix;

    public Logement() {
    }

    public Logement(int reference, String adresse, String delegation, String
gouvernorat, String type, String description, float prix) {
        this.reference = reference;
        this.adresse = adresse;
        this.delegation = delegation;
        this.gouvernorat = gouvernorat;
        this.type = type;
        this.description = description;
        this.prix = prix;
    }

    public int getReference() {
        return reference;
    }

    /*** A generer les get et set
```

- au niveau de l'entité "**Logement**", 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. Configuration de l'application Web JAX-RS :

Une application JAX-RS doit contenir au moins une classe ressource packagée dans une archive WAR. Afin d'activer les ressources REST définies dans votre application, vous devez ajouter une sous-classes javax.ws.rs.core.Application, puis ajouter les modèles d'URI requis à l'aide de l'annotation **@ApplicationPath** .

Créez une classe nommée «RestActivator» dans le package «rest.utilities» comme suit:

```
package tn.esprit.rest;

import javax.ws.rs.ApplicationPath;

@ApplicationPath("rest")
public class RestActivator extends Application{

}
```

NB: Le chemin défini dans l'annotation `javax.ws.rs.ApplicationPath` sera ajouté à l'URL de base.

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

A. Ressource Logement

1) Création d'un nouveau logement

Requête POST /logements
Content-Type: application/xml

```
<logement>
<reference>1</reference>
<adresse> 27, Rue des roses </adresse>
<gouvernorat> Ariana </gouvernorat>
<type>Studio</type>
<description> cuisine equipee </description>
<prix> 300 </prix>
<delegation> El ghazela </delegation>
</logement>
```

Réponse
Status 201Created | 404 Not Found

2) Récupération de la liste de tous les logements

Requête GET /logements

Réponse**Status** : 200 Ok**Content-Type**: application/json

```
[5]
0:
{
  reference: 1
  adresse: "27, Rue des roses"
  gouvernorat: "Ariana"
  type: "Studio"
  description: "cuisine equipee"
  prix: 300
  delegation: "El ghazela"
}
...
4:
{
  reference: 4
  adresse: "78, Rue des Oranges"
  gouvernorat: "Tunis"
  type: "EtageVilla"
  description: "chauffage central, ascenseur, climatisation"
  prix: 400
  delegation: "Bardo"
}
```

3) Récupération de la liste des logements d'une délégation spécifique

Requête GET /logements?delegation=El ghazela**Réponse****Status** : 200 Ok**Content-Type**: application/json

```
[2]
0:
{
  reference: 1
  adresse: "27, Rue des roses"
  gouvernorat: "Ariana"
  type: "Studio"
  description: "cuisine equipee"
  prix: 300
  delegation: "El ghazela"
}
-
```

4) Suppression d'un logement ayant un identifiant spécifique

Requête DELETE /logements/4**Réponse****Status** :200 Ok | 404 Not Found

5) Modification d'un logement ayant un identifiant spécifique

Requête PUT /logements/1
Content-Type: application/xml

```
<logement>
<reference>1</reference>
<adresse> 30, Rue des nilles </adresse>
<gouvernorat> Nabeul </gouvernorat>
<type>S+2</type>
<description> cuisine equipee </description>
<prix> 500 </prix>
<delegation> Nabeul </delegation>
</logement>
```

Réponse

Status :200 Ok | 404 Not Found

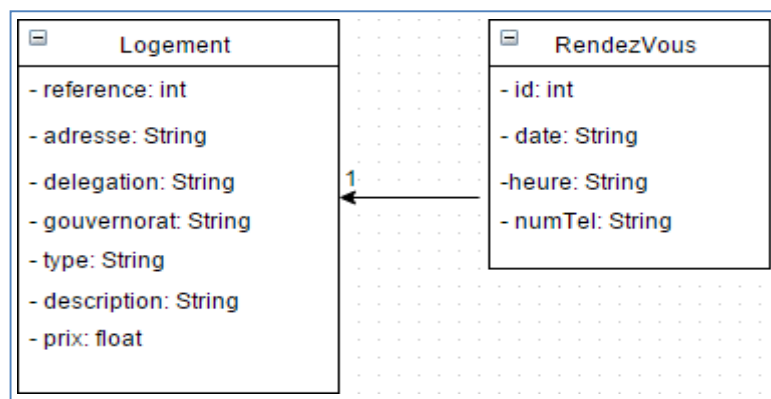
6) Récupération d'un logement ayant une référence

Requête GET /logements?reference=4

Réponse

Status : 200 Ok | 404 Not Found
Content-Type: application/json
{
 reference: 4
 adresse: "78, Rue des Oranges"
 delegation: "Bardo"
 gouvernorat: "Tunis"
 type: "EtageVilla"
 description: "chauffage central, ascenseur, climatisation"
 prix: 400
}

Pour obtenir un logement conventionné, l'étudiant peut fixer un rendez-vous pour visiter un logement parmi l'ensemble des logements disponibles.



B. Ressource Rendez-vous

1) Création d'un nouveau rendez-vous

Requête POST /rendezvous
Content-Type: application/json
{
 "id": "3",
 "date": "17-9-2015",
 "heure": "10:15",
 "logement": {"reference": "4"},
 "numTel": "55214078"
}

Réponse

Status 201Created | 404 Not Found

2) Récupération de la liste de tous les rendez-vous

Requête GET /rendesvous

Réponse

Status 200 Ok

Content-Type: application/json

```
[3]  
0: {  
  id: 1  
  date: "31-10-2015"  
  heure: "15:30"  
  logement:  
  {  
    reference: 4  
    adresse: "78, Rue des Oranges"  
    gouvernorat: "Tunis"  
    type: "EtageVilla"  
    description: "chauffage central, ascenseur, climatisation"  
    prix: 400  
    delegation: "Bardo"  
  }  
  -  
  numTel: "55214078"  
}
```

3) Récupération de la liste des rendez-vous pour la visite d'un logement ayant une référence spécifique

Requête GET /rendezvous?refLogement=4

Réponse

Status 200 Ok | 404 Not Found

Content-Type: application/json

```
[2]
0: {
  id: 1
  date: "31-10-2015"
  heure: "15:30"
  logement:
  {
    reference: 4
    adresse: "78, Rue des Oranges"
    delegation: "Bardo"
    gouvernorat: "Tunis"
    type: "EtageVilla"
    description: "chauffage central, ascenseur, climatisation"
    prix: 400
  }
  -
  numTel: "55214078"
}
-
1: {
  id: 3
  date: "17-09-2015"
  heure: "9:15"
  logement:
  {
    reference: 4
    adresse: "78, Rue des Oranges"
    delegation: "Bardo"
    gouvernorat: "Tunis"
    type: "EtageVilla"
    description: "chauffage central, ascenseur, climatisation"
    prix: 400
  }
  -
  numTel: "98102102"
}
```

4) Récupération durendez-vous ayant un identifiant spécifique

Requête GET /rendezvous/2

Réponse

Status 200 Ok | 404 Not Found

Content-Type: application/json

```
{
  id: 2
  date: "20-12-2015"
```

```
heure: "9:00"
logement:
{
reference: 1
adresse: "27, Rue des roses"
delegation: "El ghazela"
gouvernorat: "Ariana"
type: "Studio"
description: "cuisine equipee"
prix: 300
}
-
numTel: "21300811"
}
```

5) Suppression du rendez-vous ayant un identifiant spécifique

Requête DELETE /rendezvous/2
Réponse Status :200 Ok 404 Not Found

6) Modification d'un rendez-vous ayant un identifiant spécifique

Requête PUT /rendezvous/1 Content-Type : application/json { "id": "1", "date": "17-9-2015", "heure": "10:15", "logement": {"reference": "4"}, "numTel": "55214078" }
Réponse Status :200 Ok 404 Not Found