

Java – EJB (1 séance)

Raksmey PHAN

Ce TP a pour objectif de vous faire découvrir une utilisation possible du module EJB de Netbeans. A travers cet exemple, nous allons implémenter une Application Web qui se connectera à l'EJB afin de fouiller dans la base de données. Nous créerons les liens qui permettent de faire communiquer les différentes parties : Web Application -> EJBModule -> MySQL

Le schéma suivant montre les différentes parties du programme :

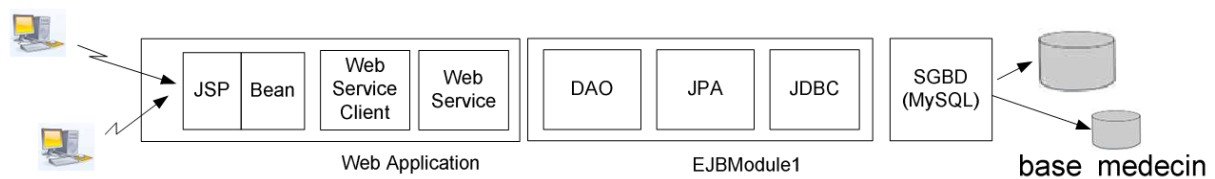


Figure 1: Différentes parties de notre web service.

I – Installation de la base de donnée (20min)

- 1) Télécharger (http://fc.isima.fr/~phan/tp/web_service/mysql-5.5.8-win32.msi) et installer-le de manière complète.
Configuration :
 - Port 3309
 - mdp : admin
- 2) Télécharger MySQL Workbench qui permet de créer facilement les tables de la base de données *base_medecin*. Installer-le complètement.
- 3) Télécharger le script (fc.isima.fr/~phan/tp/web_service/createsql.txt) qui permet de créer la table *base_medecin* avec MySQL Workbench.

I - Création d'une Web Application (40min)

- 1) Avec NetBeans 6.9.1, créer un nouveau projet dans *c:/VotreNom/*, de type *Java Web -> Web Application* qui s'appelle *WebApplicationJSP*.
Config :
 - Glassfish v3
 - Java EE 6 Web
 - Framework JavaServer Faces
- 2) Créer une connexion à la base de données *base_medecin* de NetBeans vers MySQL.
- 3) Télécharger l'EJB suivant et l'ajouter dans les librairies du projet.
http://fc.isima.fr/~phan/tp/web_service/EJBModule1_3309.jar
- 4) Ajouter un Web Service dans *Source Packages*, qui a pour nom *WSDaoJpa*, pour package *rendezvous* et qui utilise le service *DaoJpa* de l'EJB que l'on vient d'ajouter. (hint : option *Create Web Service From Existing Session Bean*).

- 5) Modifier la classe *WSDaoJpa* pour qu'il implémente l'interface *Idao* (ajouter les packages nécessaires et corriger les warnings si nécessaire). Modifier la fonction qui suit, qui permet d'obtenir la liste de tous les clients :

```
public List<Clients> getAllClients()
{
    return dao.getAllClients();
}
```

(remarque : *dao* est un objet de type *IdaoLocal*.)

- 6) Déployer le service web pour vérifier que ça marche.
- 7) A partir de l'exemple ci-dessous, écrire les méthodes permettant de :
- Lister les médecins
 - Lister les créneaux d'un médecin donné
 - Ajouter un rendez-vous
 - Supprimer un rendez-vous

II - Création de la couche JSP + Bean (1h)

- 1) De l'application précédente, ajouter un fichier de type JSP, nommé *index*.
- Ajouter le code suivant et vérifier que votre page fonctionne (clique droit sur *index* puis *run file*). Décrire le code dans le corps de la page html.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1>Hello World!</h1>

    La date courante est : <%= new java.util.Date() %>

    <%
    String[] langages = {"Java", "C++", "Smalltalk", "Simula 67"};
    out.println("<h3>Principaux langages orientés objets : </h3>");
    for (int i=0; i < langages.length; i++) {
        out.println("<p>" + langages[i] + "</p>");
    }
    %>

  </body>
</html>
```

- 2) Créer un *Package Java* dans *Source Package*, nommé *bean*. Créer à l'intérieur de ce package une *Java Class*, nommée *IndexClient*. Copier le code suivant dans votre class et décrire le code.

```
package bean ;

import java.util.ArrayList;
import java.util.List;
import rendezvous.WSDaoJpaPort;
import rendezvous_client.WSDaoJpaService;
import rendezvous_client.Clients;

public class IndexClient {

    private List<Clients> clients;
```

```

public void setClients(List<Clients> clients) {
    this.clients = clients;
}

public List<Clients> getClients()
{
    WSDaoJpaService ws = new WSDaoJpaService ();
    System.out.println("--- affichage liste des clients ---");
    List<Clients> myArr = new ArrayList<Clients>();
    myArr = ws.getWSDaoJpaPort().getAllClients();
    return myArr;
}
}

```

- 3) Modifier *index.jsp* pour ajouter le code suivant dans le *header*. Ce code vous permet d'utiliser la classe *IndexClient* pour avoir la liste de tous les clients. Décrire le code ci-dessous (corriger le code si besoin) :

```

<script language="javascript">
    function getClients()
    {
        <%
            bean.IndexClient ind = new bean.IndexClient();
            StringBuffer val = new StringBuffer();
            for (rendezvous_client.Clients c : ind.getClients()) {
                val.append(c.getNom()+" <br/> ");
            }
        %>
        var chaine = "<%= new String(val) %>";
    }
</script>

```

- 4) Modifier votre code pour ajouter un bouton et une zone texte pour afficher la réponse *chaine*, de la fonction ci-dessus.
- 5) En vous inspirant de la classe *IndexClient*, modifier votre programme pour avoir la liste de tous les médecins.