

T.P. 5 JSTL, MVC

Dans ce TP, l'utilisation des tags **JSTL** est obligatoire pour écrire les JSPs, il est interdit d'utiliser du code java dans les JSPs (c'est-à-dire, il ne faut pas utiliser des scriptlets `<%=...%>` ou `<%...%>`).
Pour le mapping des tables de la base de données, vous pouvez utiliser EJB3 ou JDBC.

L'objectif de ce TP est de développer une application Web permettant de gérer des contacts (créer un contact, rechercher un contact, modifier un contact, ...). Cette application doit être conforme au design pattern **MVC** (i.e, des JSPs pour les vues, une Servlet comme contrôleur et des classes Java (EJB3 ou JDBC) pour le modèle). Pour cela, nous allons créer des JSPs responsables de l'affichage d'un contact, de son édition,... Chacune de ces JSPs communiquera avec une Servlet qui aura le rôle de contrôleur : `ControllerServlet.java` (voir le code de cette Servlet en annexe). L'idée ici est que toutes vos requêtes doivent être sous la forme: **`ControllerServlet?do_this=[action à exécuter]`**, le paramètre **`do_this`** est récupéré par la `ControllerServlet` pour connaître l'action à effectuer (par exemple, sélectionner la JSP qui aura à afficher la réponse).

- Partie Modèle:

Vous devez commencer par créer la table Contact:

```
CREATE TABLE Contact (  
    ID_CONTACT generated always as identity primary key,  
    FIRSTNAME VARCHAR(255),  
    LASTNAME VARCHAR(255),  
    EMAIL VARCHAR(255),  
    PHONE VARCHAR(255),  
    ADDRESS VARCHAR(255)  
)
```

Créez un projet `TP5_Contacts`. Ensuite, dans un package nommé *model* sous le répertoire *src*, implémentez la classe ***Contact*** (un EJB Entity ou un simple `JavaBean` à mapper avec `JDBC`) correspondant au mapping de la table `Contact`. Implémentez également la classe ***ContactFacade.java*** dans le package *model*.

- Partie Contrôleur:

Dans un package nommé *controller*, créez une Servlet `ControllerServlet.java`. Ensuite, copiez le code de cette Servlet fournit en annexe A. Ce code est incomplet, dans la suite du TP, vous aurez à le compléter en fonction des besoins des JSPs développées dans la partie Vues.

- Partie Vues:

L'utilisateur arrive sur une page d'accueil **accueil.jsp**. Cette page affiche la liste des contacts déjà créés. En plus, la page **accueil.jsp** contient les actions suivantes (liens) :

- Créer un nouveau contact (redirection vers **addContact.jsp**)
- Supprimer un contact (redirection vers **removeContact.jsp**)
- Modifier un contact (redirection vers **updateContact.jsp**)
- Rechercher un contact (redirection vers **searchContact.jsp**)

Le code de la page **accueil.jsp** est fourni en Annexe B, copiez ce code dans **TP5_Contacts/WebContent/**, puis complétez la partie du code qui affiche le tableau des contacts. Cette partie utilise le tag JSTL `<c:forEach>` (voir cours JSTL), le but est de générer dynamiquement un tableau HTML sans utiliser une boucle «for» en code Java.

Attention: les JSPs ne peuvent pas accéder directement à la base de données car dans MVC les Vues n'appellent pas directement le Modèle. En effet, tous les appels vers le Modèle (à travers **ContactFacade.java**) doivent passer par le Contrôleur **ControllerServlet.java**. Par exemple, pour accéder à la page d'accueil **accueil.jsp**, l'utilisateur doit appeler la **ControllerServlet** (URL: http://localhost:8080/TP5_Contacts/ControllerServlet). Ensuite, la **ControllerServlet** charge la liste des contacts dans la requête (`request.setAttribute("listContacts", contactFacade.findAll())`), puis, elle transmet la requête vers **accueil.jsp** (`forward(request, response);`). Enfin, **accueil.jsp** récupère la liste des contacts à partir de la requête et l'affiche (`<c:forEach items="${requestScope.listContacts}">`).

La page **addContact.jsp** propose un formulaire qui contient des champs pour renseigner un contact. Ses champs doivent correspondre aux attributs de la classe **Contact.java**. Le formulaire doit également avoir un bouton «submit».

Une fois le formulaire envoyé, il est traité par la servlet **ControllerServlet**. Par conséquent, afin de préciser à **ControllerServlet** l'action à effectuer, le formulaire de **addContact.jsp** doit contenir un paramètre caché: `<input type="hidden" name="do_this" value="create"/>` (voir le code du formulaire ci-dessous). Créez la page **addContact.jsp** dans **TP5_Contacts/WebContent/**, puis copiez et complétez le code du formulaire suivant:

```
<form action="ControllerServlet" method="POST">
<input type="hidden" name="do_this" value="create" />
<table>
<tr>
<td align="center" colspan="2"><font size="4">Please Enter the Following Details</font>
</td>
</tr>
<tr><td>First Name</td><td><input type="text" name="firstName" /></td></tr>
<tr><td>Last Name</td><td><input type="text" name="lastName" /><br/></td></tr>
...
...
<tr><td></td><td><input type="submit" name="Validate" value="save" /></td></tr>
</table>
</form>
```

Ensuite, la Servlet **ControllerServlet** récupère les informations du contact à partir des données du formulaire puis elle utilise le **ContactFacade** pour appeler la méthode **addContact(long id, String firstname, String lastname, String email, ...)**. Cette méthode crée une instance de **Contact**, fixe ses attributs puis le sauvegarde dans la base de données.

Même principe pour **removeContact.jsp**. Cette page propose un formulaire avec un champ nommé **idContact** et un bouton «submit». Une fois cliqué, le formulaire fait appel à la servlet **ControllerServlet** qui récupère l'id du contact à partir des données du formulaire puis elle appelle la

méthode ***deleteContact(long id)*** de ***ContactFacade***. Cette méthode se charge d'effacer le contact portant cet id dans la base de données.

Ainsi de suite pour les autres fonctionnalités.

Annexe A. Servlet contrôleur *ControllerServlet.java*:

```
package controller;

import java.io.IOException;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletConfig;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import model.ContactFacade;

public class ControllerServlet extends HttpServlet {

    public ControllerServlet() {
        super();
    }

    public void init(ServletConfig config) throws ServletException {
        System.out.println("*** initializing controller servlet.");
        super.init(config);
    }

    protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        doPost(request, response);
    }

    protected void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {

        // recuperation de l'action à effectuer
        String do_this = request.getParameter("do_this");
        if (do_this == null) {
            // definir le contexte pour une redirection sur la page accueil.jsp
            ServletContext sc = getServletContext();
            RequestDispatcher rd = sc.getRequestDispatcher("/accueil.jsp");
            // charger la liste des contacts dans la requête pour les
            // transmettre à la JSP accueil.jsp (qui va les afficher)
            request.setAttribute("listContacts", contactFacade.findAll());
            rd.forward(request, response);

        } else if (do_this.equals("delete")) {
            // recuperation de l'id du contact
            String id = (String) request.getParameter("contact_id");
            if (id == null) {
                // redirection sur la page removeContact.jsp
                response.sendRedirect("removeContact.jsp");
            } else {
                // id non nul, donc on supprime le contact identifié par id
                contactFacade.deleteContact(id);
                // on recharge la page d'accueil
                response.sendRedirect("ControllerServlet");
            }
        } else if (do_this.equals("create")) {
            // recuperation du nom contact
            String lastName = (String) request.getParameter("lastName");
            if (lastName == null) {
                // redirection sur la page removeContact.jsp
                response.sendRedirect("addContact.jsp");
            } else {
                // TODO le nom n'est pas nul, donc on ajoute le contact dans la base
                contactFacade.createContact(...);
                // on recharge la page d'accueil
                response.sendRedirect("ControllerServlet");
            }
        }
    }
}
```

```

        }

        } else if (do_this.equals("update")) {
            // TODO ...
        } else if (do_this.equals("search")) {
            // TODO ...
        }
        // TODO ...
    }
}

```

Annexe B. page d'accueil *accueil.jsp*:

```

<%@page import="model.entity.Contact"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!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=ISO-8859-1">
<title>Accueil</title>
</head>
<body>
    <h1>Liste des contacts : </h1>
    <table border="2">
        <tr>
            <th>Contact Id</th>
            <th>First Name</th>
            <th>Last Name</th>
            <th>Email</th>
            <th>Phone Number</th>
            <th>Address</th>
        </tr>
        <c:forEach var="contact" items="${requestScope.listContacts}">
            <tr>
                <td>...</td>
                <td>...</td>
                <td>...</td>
                <td>...</td>
                <td>...</td>
                <td>...</td>
            </tr>
        </c:forEach>
    </table>
    <a href="ControllerServlet?do_this=create" style="">Créer un nouveau contact</a>
    <a href="ControllerServlet?do_this=delete" style="">Supprimer un Contact</a>
    <a href="ControllerServlet?do_this=update" style="">Modifier un Contact</a>
    <a href="ControllerServlet?do_this=search" style="">Rechercher un contact</a>
</body>
</html>

```