ESIEA 4A

Module CAAOO
Responsable : Michel Futtersack

Travaux pratiques JEE

Exercice 1

Télécharger le projet `MaSaison` à l'adresse :

http://www.droit.univ-paris5.fr/futtersack/divers/MaSaison.rar

Décompresser l'archive (avec le programme 7-zip par exemple). Lancer NetBeans et ouvrir le projet `MaSaison`, puis l'exécuter (F6)

a) Ajouter dans la page `index.jsp` une liste déroulante (dans le même formulaire que celui qui existe) permettant de choisir une voyelle : A, E, I, O ou U. Créer un Bean nommé Rimbaud (dans le panneau Projects en haut à gauche, clic droit sur le nom du projet `MaSaison -> New -> Java Class`) dans le package `mesbeans`. Ce Bean possède une propriété `voyelle` de type `String` dont la valeur est initialisée par la requête HTTP et une propriété `couleur` de type `String` dont la valeur sera calculée suivant la voyelle choisie par l'utilisateur :

A-> noir, E-> blanc, I-> rouge, U-> vert, O-> bleu

Modifier la page `MaSaison.jsp` de façon qu'elle affiche en plus le nom de la couleur correspondant à la voyelle choisie par l'utilisateur.

b) Renommez la page `index.jsp` en `choix.jsp` (dans le panneau Projects, dérouler l'arbre du projet `MaSaison` puis le nœud `Web Pages`. Faites un clic droit sur `index.jsp ->Rename`…). Créer une nouvelle page `index.jsp` permettant de saisir un login (conseil : créer cette page « à la souris » avec un éditeur HTML et recopiez le code HTML généré dans votre page `index.jsp`). Les données saisies dans le formulaire de cette page sont envoyées à une nouvelle page nommée `login.jsp`. Nous allons créer un Bean nommé `Login` contenant deux propriétés de type `String` nommées `identifiant` et `motDePasse` et une propriété de type `boolean` nommée `autorisé`. Cette classe `Login` contient également un attribut privé nommé `listeLogins`, qui contient la liste des couples (identifiant, mot de passe) autorisés. La méthode `getAutorisé` de la classe `Login` vérifie si les valeurs de l'identifiant et du mot de passe correspondent à un couple autorisé. Voici le code de la classe `Login` :

```
package mesBeans;
import java.util.*;
public class Login {
   private Map listeLogins;
   private String identifiant;
   private String motDePasse;
   private boolean autorisé;
   public Login(){
      listeLogins = new HashMap();
      listeLogins.put("toto","toto");
      listeLogins.put("titi","titi");
 }
public void setIdentifiant(String valeur) { identifiant = valeur; }
public  String getIdentifiant() {return identifiant;}
public void setMotDePasse(String valeur) { motDePasse = valeur; }
public  String getMotDePasse() {return motDePasse;}
```

```
  public void setAutorisé(boolean valeur) { autorisé = valeur; }
  public  boolean getAutorisé() {
    return(motDePasse.equals(listeLogins.get(identifiant)));
    }
}
```

La page `login.jsp` utilise le Bean `Login` et redirige (`<jsp:forward…>`) vers la page `choix.jsp` si la valeur de la propriété `autorisé` est `true`, sinon elle redirige vers la page `index.jsp`.

c) Fermer votre navigateur Web puis relancer l'application Web `MaSaison`. Essayer d'aller directement sur la page choix.jsp en complétant dans la barre de navigation l'URL : http://localhost:8084/MaSaison/choix.jsp

Comment résoudre ce GROS problème de sécurité ? On ne peut pas utiliser le bean `Login` pour tester la valeur de son attribut `autorisé` car si l'utilisateur va directement à la page `choix.jsp`, le bean `Login` n'a pas été encore instancié dans la session. On va utiliser l'objet prédéfini `session` en lui définissant un attribut `connecté` avec la valeur `OK` dans le cas où le login de l'utilisateur est correct. Voici le nouveau code de la page `login.jsp` :

```
<html><head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<jsp:useBean id="log" class="mesBeans.Login" scope="session"/>
<jsp:setProperty name="log" property="identifiant"/>
<jsp:setProperty name="log" property="motDePasse"/>
</head>
<body>
<% if (log.getAutorisé()){
     session.setAttribute("connecté","OK"); %>
    <jsp:forward page="choix.jsp" />
<% } %>
<jsp:forward page="index.jsp" />
</body>
</html>
```

Cette JSP ne renvoie aucun code HTML au navigateur. Son rôle est d'indiquer quelle est la prochaine JSP à appeler suivant les valeurs entrées par l'utilisateur dans le formulaire de `index.jsp`. A quelle architecture vous fait penser le rôle de `login.jsp` ?

Comment faut-il modifier la page `choix.jsp` pour que si l'utilisateur y accède directement, il soit redirigé vers la page `index.jsp` ?

Exercice 2

Le but de cet exercice est de réaliser un client Web léger de la base ACCESS ComptoirPlus

Télécharger cette base de données à l'adresse :
http://www.droit.univ-paris5.fr/futtersack/divers/ComptoirPlus.rar
Créer un nouveau projet NetBeans nommé ClientComptoir, de type Web Application
a) remplacer la page `index.jsp` par la page `index.jsp` du projet MaSaison de l'exercice précédent (dans le menu File de NetBeans choisir Open File…et copier/coller le source). Créer une page `login.jsp` identique à celle du projet MaSaison et créer un bean `Login` identique au Bean `Login` du projet MaSaison. Nous voulons que la vérification du login se fasse maintenant en utilisant la table Employés de ComptoirPlus. Pour cela, il suffit de supprimer l'attribut `listeLogins` de type `Map`, ainsi que le constructeur `Login()` permettant d'initialiser cet attribut, puis de remplacer le code de la méthode `getAutorisé()` par :

```
public boolean getAutorisé() {
  boolean résultat = false;
  Connection conn = null;
  try {
```

```
    Class.forName ("sun.jdbc.odbc.JdbcOdbcDriver");
    conn = DriverManager.getConnection ("jdbc:odbc:DRIVER=Microsoft Access
Driver (*.mdb); "
    + "DBQ=ComptoirPlus; " + "DefaultDir=C:\\Documents and Settings\\le
BOSS\\Bureau\\TP J2EE\\ClientComptoir ");
    Statement s = conn.createStatement ( );
    s.executeQuery (" SELECT * FROM Employés WHERE Nom = ‘ " + identifiant
+ " ’ AND [Mot de passe] = ‘ " + motDePasse + " ’ ");
    ResultSet rs = s.getResultSet ( );
    if(rs.next()) résultat=true;
    rs.close(); s.close();conn.close();
  }
  catch (Exception e) {System.err.println (" Pb connexion ComptoirPlus ");}
  return résultat;
}
```

On n’oubliera pas d’ajouter l’instruction :
`import java.sql.*;`
en début de fichier pour pouvoir utiliser les noms courts de JDBC et on changera également le chemin d’accès à la base de données ComptoirPlus dans la chaîne de connexion.
Dans le cas où l’utilisateur est authentifié, le contrôleur (au sens MVC) `login.jsp` redirige vers une page `choixClient.jsp` qui pour le moment se contente d’afficher "Vous êtes reconnu ", sinon il redirige vers `index.jsp`.

b) Créer la page `saisirLogin.jsp` qui permet d’entrer un nouveau login et qui envoie celui-ci à une page `nouveauLogin.jsp` qui mémorise ce nouveau login dans la BD et redirige vers la page `choixClient.jsp`. La page `saisirLogin.jsp` est appelable par un hyperlien placé dans la page `choixClient.jsp`.

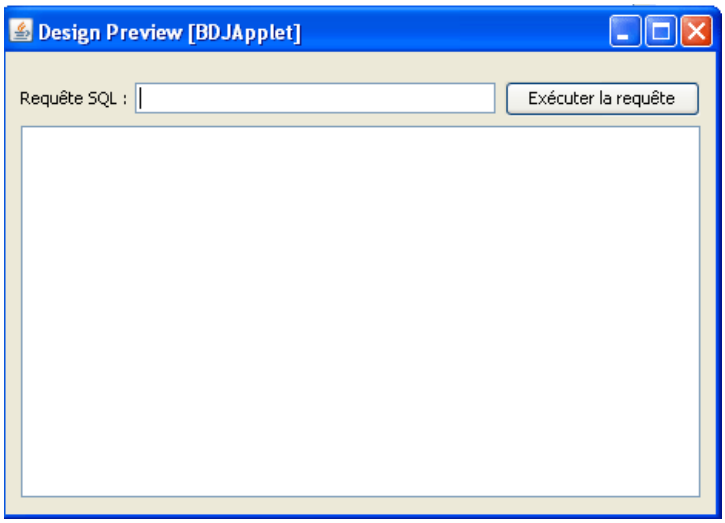
c) Créer la page `choixClient.jsp` qui affiche la liste des noms des clients de ComptoirPlus dans une liste déroulante. (élément HTML `<select>`. Pour cela on créera un Bean `Clients` contenant un attribut de type `List <String>` nommé `nomsClients`. Pour que le compilateur accepte les types génériques, aller dans les propriétés du projet (clic droit sur le nom du projet dans le panneau Projects) et dans la catégorie Sources, choisissez en bas à droite Source level : 1.5. La propriété `nomsClients` sera initialisée dans le constructeur du Bean Clients par une requête à la base de données. On utilisera une instance de `Clients` dans `choixClient.jsp` (balise <jsp:useBean>) et on générera les items du `<select>` à l’intérieur d’une scriptlet. La valeur de l’attribut `name` de l’élément `<select>` sera `nomClient`. La valeur de l’attribut action de l’élément `<form>` sera `afficheClient.jsp`.

d) Créer la page `afficheClient.jsp` dans laquelle on affiche toutes les informations concernant le client choisi par l’utilisateur.

Exercice 3

Le but de cet exercice est de réaliser une application Web à 3 niveaux avec une interface riche sous la forme d'une applet communiquant avec une servlet, celle-ci communiquant avec la BD ComptoirPlus. Pour cela nous allons réaliser deux projets NetBeans. Le premier pour réaliser l'applet et le deuxième pour réaliser l'application Web dans laquelle nous intégrerons l'applet.

Créer un nouveau projet NetBeans nommé ComptoirPlusApplet de type General->Java Application. Cliquer-droit sur le nom du projet, sélectionner New File/Folder -> Java GUI Forms ->JApplet Form. Nommez cette applet BDJApplet et placez-la dans un package nommé comptoirplus. Dessiner l'IHM suivante en mode Design (si la palette d'outils n'est pas visible, appelez-la à partir du menu Window de NetBeans) :



Cette IHM contient un `JLabel`, un `JTextField`, un `JButton` et un `JTextArea`.

Pour écrire le code du gestionnaire de l'événement Action émis par le bouton : Cliquer-droit sur le bouton ->Events->Action->actionPerformed et saisir le code suivant :

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
String requête = jTextField1.getText();
try {
 java.net.URL url = new
java.net.URL("http://localhost:8084/ComptoirPlusRIA/BDServlet?requete=" +
java.net.URLEncoder.encode(requête,"UTF-8"));
  java.net.URLConnection uc = url.openConnection();
  uc.setDoInput(true);
  uc.setDoOutput(true);
  uc.setUseCaches(false);
  uc.setRequestProperty("Content-type","application/x-www-form-
urlencoded");
  java.io.InputStreamReader in = new
java.io.InputStreamReader(uc.getInputStream());
  int chr = in.read();
  while (chr != -1) {
    jTextArea1.append(String.valueOf((char) chr));
    chr = in.read();
  }
  in.close();
  }
catch(java.net.MalformedURLException e) {
  jTextArea1.setText(e.toString());}
catch(java.io.IOException e){
  jTextArea1.setText(e.toString());}
}
```

Créer un nouveau projet NetBeans nommé ComptoirPlusRIA de type Web->Web Application. Cliquer-droir sur le nom de ce projet -> New -> Servlet. Nommez BDServlet cette nouvelle servlet et placez-la dans le package comptoirplus.

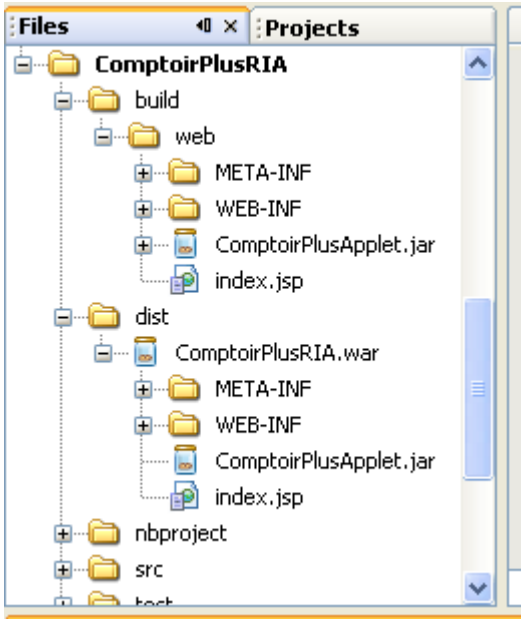
Remplacez le corps de la méthode `processRequest` par le suivant (ne pas oublier d'ajouter `import java.sql.*;` en début de fichier et de modifier le chemin d'accès à la base ComptoirPlus) :

```
response.setContentType("text/html;charset=iso-8859-1");
PrintWriter out = response.getWriter();
```

```
try {
  Class.forName ("sun.jdbc.odbc.JdbcOdbcDriver");
  Connection conn = DriverManager.getConnection
("jdbc:odbc:DRIVER=Microsoft Access Driver (*.mdb); "
+"DBQ=ComptoirPlus;"+"DefaultDir=C:\\Documents and Settings\\le
BOSS\\Bureau\\TP J2EE\\ClientComptoir");
  System.out.println ("Connexion etablie avec COMPTOIRPLUS");
  Statement s = conn.createStatement ( );
  s.executeQuery ("SELECT [Code client] FROM Clients");
  ResultSet rs = s.getResultSet ( );
  int nbColonnes = rs.getMetaData().getColumnCount();
  while (rs.next ()){
    for(int i=1; i<=nbColonnes; i++) out.print(rs.getString(i) + " ");
    out.println("");
  }
rs.close (); s.close ( ); conn.close();
}
catch (Exception e) {System.err.println ("Pb de connexion à
ComptoirPlus");}
out.close();
```

Nous allons maintenant intégrer l'applet dans notre application web. Cliquer-droit sur le nom du projet ComptoirPlusRIA->Properties->Packaging (dans le panneau gauche)->Add Project. Indiquez le projet ComptoirPlusApplet.

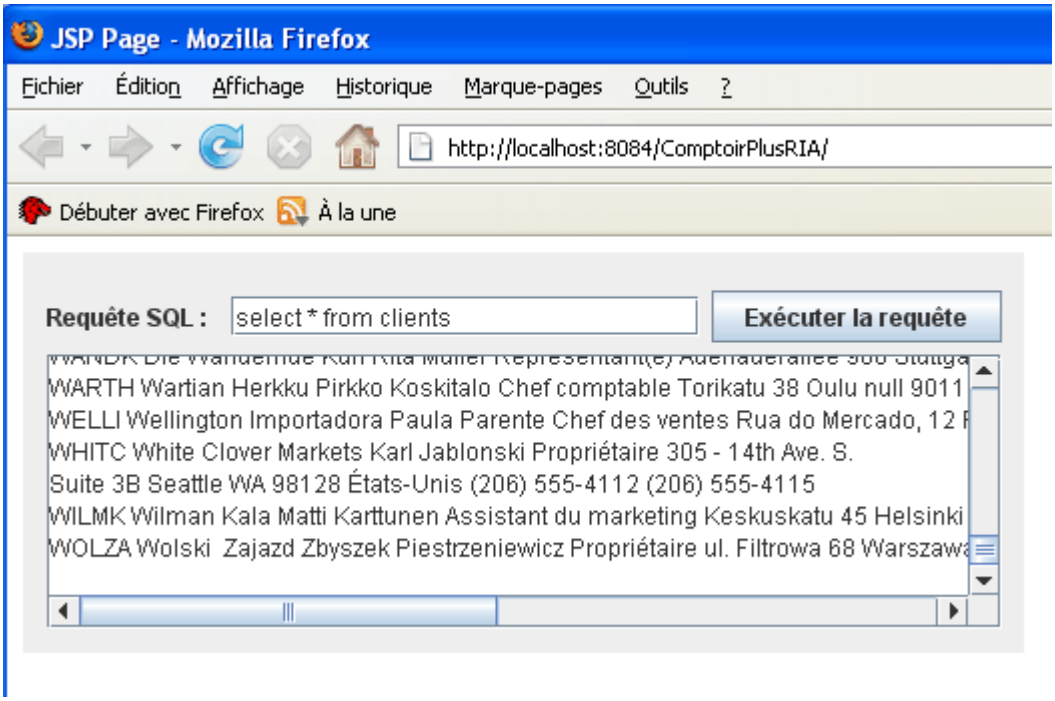
Construire (Build) l'application Web. Ouvrez le panneau Files du projet ComptoirPlusRIA (Ctrl+2) : vous devez voir l'arborescence suivante :



Remplacer le corps de la page `index.jsp` par :

```
<body>
  <applet code="comptoirplus.BDJApplet" archive="ComptoirPlusApplet.jar"
width=500 />
</body>
```

Testez votre application Web en entrant par exemple la requête "select * from clients" dans la zone de saisie de l'applet :


JSP Page - Mozilla Firefox
Fichier Édition Affichage Historique Marque-pages Outils ?
http://localhost:8084/ComptoirPlusRIA/
Débuter avec Firefox À la une
Requête SQL :
select * from clients
Exécuter la requête
WARTH Wartian Herkku Pirkko Koskitalo Chef comptable Torikatu 38 Oulu null 9011
WELLI Wellington Importadora Paula Parente Chef des ventes Rua do Mercado, 12
WHITC White Clover Markets Karl Jablonski Propriétaire 305 - 14th Ave. S.
Suite 3B Seattle WA 98128 États-Unis (206) 555-4112 (206) 555-4115
WILMK Wilman Kala Matti Karttunen Assistant du marketing Keskuskatu 45 Helsinki
WOLZA Wolski Zajazd Zbyszek Piestrzeniewicz Propriétaire ul. Filtrowa 68 Warszawa