

CONTRÔLE FINAL DÉVELOPPEMENT N-TIERS J2EE

Questions de cours (5 pts)

1. Schématisez l'architecture 3-tiers RPC ?
2. Que veut dire : déployer un projet dans un serveur d'application ?
3. Donnez les 3 types de composants EJB.
4. Quel est le type d'EJB qui n'accepte de recevoir qu'un seul client à la fois ?
5. Dans le fichier « web.xml » d'un projet J2EE on utilise la balise <servlet-mapping>. pourquoi ?

Problème (15 pts)

Soit une base de données « INSCRIPTION » contenant la table « Etudiant » avec les attributs (id, nom, prénom, date_naiss, filiere).

Et soit le projet J2EE créé dans Eclipse appelé « Exam_2015_2016 » contenant le module Web.

1. Créer une classe appelée **DBManager.java** qui permet de gérer la base de données. Cette classe contient deux méthodes :
 - une méthode de connexion à la base `getConnection()`, on suppose qu'elle existe.
 - une méthode qui récupère les données de la table Etudiant `getEtudiants()`.
2. Créer une servlet de contrôle « liste_etudiant_servlet.java » qui fait le lien entre l'utilisateur et la vue. L'utilisateur accède à la servlet par l'url « /insc_etud », et la servlet transmet la requête à la vue « liste_etudiants.jsp ».
 - a. Donnez le contenu du fichier « web.xml » ;
 - b. Donnez le contenu de la classe « liste_etudiant_servlet.java » (`doGet()` et `doPost()`) ;
3. Dans le fichier « liste_etudiants.jsp » de la vue, montrer comment on récupère les données de la table « Etudiant » pour les afficher.

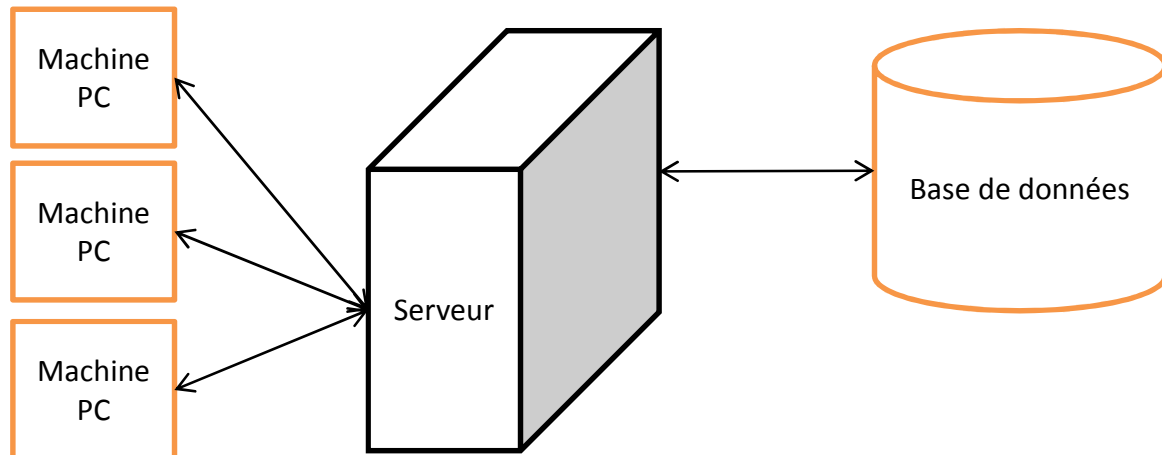
Bonne chance...

NB: Le corrigé type vous le trouverez sur le site :
http://fac-sciences.univ-batna.dz/cs/enseignants/guezouli_larbi_site/

CORRECTION DU CONTRÔLE FINAL DÉVELOPPEMENT N-TIERS J2EE

Questions de cours (5 pts)

1. Schématisez l'architecture 3-tiers RPC ?



2. Que veut dire : déployer un projet dans un serveur d'application ?

Déployer un projet dans un serveur d'application, est de le placer dans le serveur pour pouvoir l'exécuter.

3. Donnez les 3 types de composants EJB.

- Session (stateful, stateless)
- Entité (BMP, CMP)
- Message

4. Quel est le type d'EJB qui n'accepte de recevoir qu'un seul client à la fois ?

L'EJB **session** ne peut recevoir qu'un seul client à un moment donné.

5. Dans le fichier « web.xml » d'un projet J2EE on utilise la balise <servlet-mapping>. pourquoi ?

La balise <servlet-mapping> permet d'associer une servlet à une URL.

Problème (15 pts)

Soit une base de données « INSCRIPTION » contenant la table « Etudiant » avec les attributs (id, nom, prénom, date_naiss, filiere).

Et soit le projet J2EE créé dans Eclipse appelé « Exam_2015_2016 » contenant le module Web.

1. Créer une classe appelée **DBManager.java** qui permet de gérer la base de données. Cette classe contient deux méthodes :

- une méthode de connexion à la base `getConnection()`, on suppose qu'elle existe.
- une méthode qui récupère les données de la table `Etudiant` `getEtudiants()`.

```

public class DBManager { (0.5 pts)
    public static Connection getConnection() {
        Connection con = null;

        try {
            Context context = new InitialContext();
            DataSource ds = (DataSource)
context.lookup("java:jboss/datasources/bdav_exam");
            con = ds.getConnection();
        } catch (NamingException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return con;
    }
    public static ArrayList<Etudiant> getEtudiants() { (1 pt)
        ArrayList<Etudiant> etudiants = new ArrayList<Etudiant>();
        Connection con = getConnection(); (0.5 pts)
        if (con == null) {
            System.out.println("Problème de connexion à la base");
        } else {
            try {
                String myRequest = "";
                myRequest = "SELECT * FROM etudiant"; (0.5 pts)
                PreparedStatement pStmt =
con.prepareStatement(myRequest);
                ResultSet rs = pStmt.executeQuery(); (1 pt)
                int j = 0;
                while (rs.next()) { (0.5 pts)
                    Etudiant etudiant = new Etudiant(); (0.5 pts)
                    etudiant.nom = rs.getString("nom"); (0.25 pts)
                    etudiant.prenom = rs.getString("prenom"); (0.25 pts)
                    etudiant.date_nais =
rs.getDate("date_nais").toString(); (0.25 pts)
                    etudiant.filiere = rs.getString("filiere"); (0.25 pts)

                    etudiants.add(j, etudiant); (0.5 pts)
                    j++;
                }
            } catch (SQLException e) {
                e.printStackTrace();
            } finally {
                if (con != null) {
                    try {
                        con.close();
                    } catch (SQLException e) {
                        e.printStackTrace();
                    }
                }
            }
        }
        return etudiants; (0.5 pts)
    }
}

```

```
}
```

2. Créer une servlet de contrôle « liste_etudiant_servlet.java » qui fait le lien entre l'utilisateur et la vue. L'utilisateur accède à la servlet par l'url « /insc_etud », et la servlet transmet la requête à la vue « liste_etudiants.jsp ».
- Donnez le contenu du fichier « web.xml » ;

```
<web-app> (0.5 pts)
  <servlet> (0.5 pts)
    <servlet-name>liste_etudiant_servlet</servlet-name> (0.5 pts)
    <servlet-class>org.liste_etudiant_servlet</servlet-class> (0.5 pts)
  </servlet>
  <servlet-mapping> (0.5 pts)
    <servlet-name>liste_etudiant_servlet</servlet-name> (0.5 pts)
    <url-pattern>/insc_etud</url-pattern> (0.5 pts)
  </servlet-mapping>
</web-app>
```

- Donnez le contenu de la classe « liste_etudiant_servlet.java » (doGet () et doPost ()) ;

```
public class liste_etudiant_servlet extends HttpServlet { (1 pt)
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException { (0.5 pts)
        ArrayList<Etudiant> etudiants = DBManager.getEtudiants(); (0.5 pts)
        if (etudiants != null) {
            request.setAttribute("var1", etudiants); (0.5 pts)
        }
        this.getRequestDispatcher("/WEB-
INF/liste_etudiants.jsp").forward(request, response); (0.5 pts)
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException { (0.5 pts)
    }
}
```

3. Dans le fichier « liste_etudiants.jsp » de la vue, montrer comment on récupère les données de la table « Etudiant » pour les afficher.

```
<%
ArrayList<Etudiant> etudiants =
(ArrayList<Etudiant>)request.getAttribute("var1"); (1 pt)
Etudiant etd = null;
if (etudiants != null) {
    for (int i=0; i<etudiants.size(); i++) (0.5 pts)
    {
        etd = (Etudiant)(etudiants.get(i)); (0.5 pts)
    }
%>
    <tr>
        <th> <%=etd.nom%> </th>
        <th> <%=etd.prenom%> </th>
        <th> <%=etd.date_nais%> </th>
        <th> <%=etd.filiere%> </th>
    </tr>
<%
}
}%>
```