

Tutoriel 7: Les bases de données- Les requêtes

L'objectif de ce tutoriel est de vous montrer via des exemples comment se connecter à une base de données et effectuer une requête e sélection. Les deux premières étapes (étape 0 et étape 1) sont les mêmes que les deux premières étapes vues dans le tutoriel précédent (tutoriel 6). Nous allons travailler avec la même base de données (*bibliotheque*) que nous avons vue dans le cours de programmation Java dans le premier semestre.

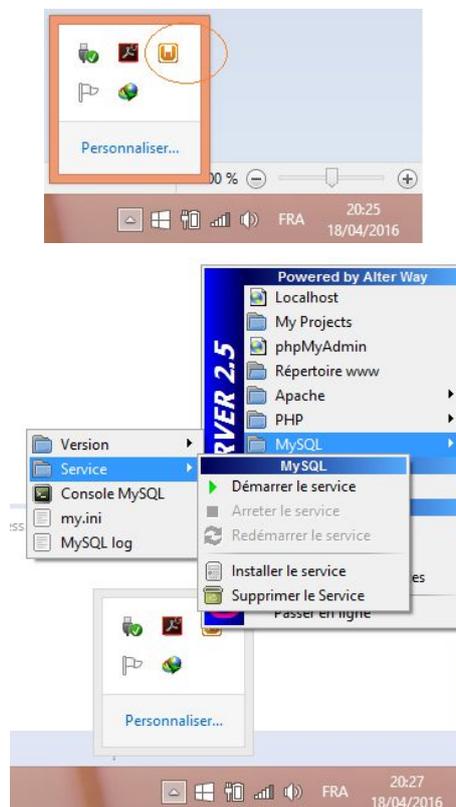
Etape 0: Création d'un nouveau projet

Si on n'a pas un projet web dynamique, on le crée selon les mêmes étapes vues dans le premier tutoriel, puis on ajoute ce projet au serveur Tomcat.

RQ: Il ne faut pas oublier de modifier le *Build Path*.

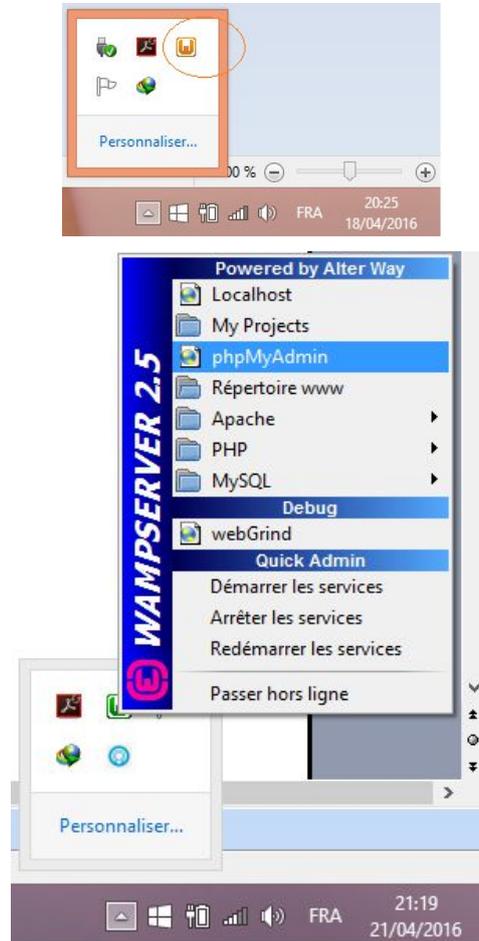
Etape1: Préparation du JDBC

- 1) Télécharger et installer MySQL, (Il suffit de télécharger et d'installer wampserver <http://www.wampserver.com/>).
- 2) Télécharger le driver (un fichier jar) : <https://dev.mysql.com/downloads/connector/j/>
- 3) Mettre le fichier jar dans le dossier *lib* qui est un sous dossier du dossier WEB-INF.
- 4) Démarrer le serveur MySQL: Cliquer sur l'icône du *wampserver*, puis sur *MySQL*, puis sur *Service*, puis sur *Démarrer le Service*.



Etape2: Création d'une base de données

1) Cliquer sur l'icône du *wampserver*, puis sur *phpMyAdmin*.



2) Si vous n'avez pas la base de données (*bibliothèque*), il faut la créer en utilisant *phpMyAdmin*, puis créer la table base de données appelée *bibliothèque*, puis créer une table appelée *livres*.



3) insérer 2 ou trois livres dans la tables *livres* (voire la figure ci-dessous).

	code	titre	auteur	nombreDePages	prix
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	1	La bête humaine	Emile Zola	170	25
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	2	Notre Dame de Paris	Victor Hugo	150	80
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	3	Pattern in Java	Mark Grand	470	150
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	5	LE RÊVE	Emile Zola	225	60

↑ Tout cocher Pour la sélection : Modifier Effacer Exporter

Nombre de lignes : 25 ▼

Etape3: Création d'une classe Java pour établir la connexion à la base et effectuer la requête

1) Créer une classe Java appelée *RequestTest* dans un package appelé *bean* par exemple (le package *bean* doit être créé dans *src*).

2) Mettre le code suivant dans la classe *RequestTest*.

```
public class RequestTest {  
  
    public ArrayList<String> executeQuery(String query) {  
  
        ArrayList<String> res = new ArrayList<String>();  
  
        Connection connection = null;  
        Statement statement = null;  
        ResultSet resultSet = null;  
  
        String url = "jdbc:mysql://localhost/bibliotheque";  
        String user = "root";  
        String password = "";  
  
        try {  
  
            Class.forName("com.mysql.jdbc.Driver");  
  
            connection = DriverManager.getConnection(url, user, password);  
  
            //System.out.println("connexion avec succès");  
  
            statement = connection.createStatement();  
  
            resultSet = statement.executeQuery(query);  
  
            while (resultSet.next()) {  
                //System.out.println(resultSet.getString(1)+",  
                "+resultSet.getString(2)+", "+resultSet.getInt(3));  
                String str = resultSet.getString(1)+",  
                "+resultSet.getString(2)+", "+resultSet.getInt(3);  
                res.add(str);  
            }  
  
            resultSet.close();  
  
        }  
  
    }  
  
}
```

```

        connection.close();

    } catch (SQLException ex) {
        System.out.println("0000");
        Logger lgr = Logger.getLogger(RequestTest.class.getName());
        lgr.log(Level.SEVERE, ex.getMessage(), ex);

    } catch (ClassNotFoundException e) {
        System.err.println("Driver loading error : " + e);
    }

    return res;
}
}
}

```

Etape4: Création du Servlet

1) Créer une classe Java appelée *ServletRequest* dans un package appelé *servlets* par exemple (le package *servlets* doit être créé dans *src*).

2) Remplacer la méthode *doGet* de classe *ServletRequest* par le code suivant:

```

protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    /* Création du bean */
    RequestTest req = new RequestTest();
    //Créer une requete
    String query = "SELECT titre, auteur, nombreDePages FROM livres";

    //Essayer de se connecter à la base de données et exécuter la requete
    ArrayList<String> res = req.executeQuery(query);

    request.setAttribute( "result", res);
    this.getServletContext().getRequestDispatcher( "/WEB-
INF/response.jsp").forward( request, response );
}

```

3) Ajouter le code suivant au fichier web.xml

```

<servlet>
    <servlet-name>Test</servlet-name>
    <servlet-class>servlets.ServletRequest</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>Test</servlet-name>
    <url-pattern>/home</url-pattern>
</servlet-mapping>

```

Etape5: Création de deux JSPs

1) Créer dans le répertoire WEB-INF, une JSP appelée *response.jsp*. Cette JSP contient le code suivant:

```

<%@ page pageEncoding="UTF-8" %>
<%@ page import="java.util.ArrayList"%>
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Test</title>
  </head>
  <body>
    <p>Connexion avec succès</p>
    <p>Voici la liste des livres</p>
    <p>
      <%
        ArrayList<String> liste = (
ArrayList<String>) request.getAttribute("result");
        for(int i = 0; i < liste.size(); i++) {

          out.println(liste.get(i)+"      ");

        }
      %>
    </p>
  </body>
</html>

```

Etape6: Exécution

Si tout va bien, on obtient le résultat ci-dessous. Pour cela, vous allez directement exécuter la *Servlet* à partir d'eclipse ou à partir d'un navigateur, e, utilisant l'url suivant :

<http://localhost:8081/testJDBC/home>

