La configuration   
et l’exploitation d’un pool   
de connexions

# Démonstration 1 du module 4

|  |
| --- |
| Cette démonstration présente la mise en place d’un pool de connexions. |

Déroulement

* Copier le pilote sqljdbc42.jar dans le répertoire /WebContent/lib.
* Créer le fichier context.xml dans le répertoire /WebContent/context.xml
* Modifier le contenu de ce fichier avec la configuration suivante :

<?xml version="1.0" encoding="UTF-8"?>

<Context>

<Resource

name="jdbc/pool\_cnx"

driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver"

type="javax.sql.DataSource"

url="jdbc:sqlserver://localhost;databasename=BDD\_DEMO"

username="utilisateurBDD"

password="Pa$$w0rd"

maxTotal="100"

maxIdle="30"

maxWaitMillis="10000"

/>

</Context>

* Créer une servlet nommée ServletTestPoolConnexion dans le package fr.eni.javaee.module4.
* Modifier le code généré par défaut par l’IDE :
  + Modifier l’URL de l’annotation @WebServlet avec la valeur suivante :

@WebServlet("/modules/module2/ServletTestPoolConnexion")

* + Modifier le code de la méthode doGet avec :

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

PrintWriter out = response.getWriter();

**try** {

Context context = **new** InitialContext();

//Recherche de la DataSource

DataSource dataSource =

(DataSource)context.lookup("java:comp/env/jdbc/pool\_cnx");

//Demande une connexion. La méthode getConnection met la demande en

attente tant qu'il n'y a

//pas de connexions disponibles dans le pool.

Connection cnx = dataSource.getConnection();

out.print("La connexion est " +

(cnx.isClosed()?"fermée":"ouverte")+".");

//Libérer la connexion lorsque l'on en a plus besoin:

cnx.close();//La connexion n'est pas fermée mais remise dans le pool.

} **catch** (NamingException | SQLException e) {

e.printStackTrace();

response.setStatus(HttpServletResponse.***SC\_INTERNAL\_SERVER\_ERROR***);

out.println("Une erreur est survenue lors de l'utilisation de la base de données : " + e.getMessage());

}

}

* Pour que l’exemple fonctionne, il faut créer une base de données nommée BDD\_DEMO qui doit être accessible par un utilisateur nommé utilisateurBDD (dont le mot de passe est Pa$$w0rd).
* Créer au niveau de cette base de données la table suivante :

CREATE TABLE AVIS

(

identifiant int IDENTITY(1,1) CONSTRAINT PK\_AVIS PRIMARY KEY,

description varchar(150) NOT NULL,

note tinyint NOT NULL CONSTRAINT CK\_AVIS\_note CHECK(note between 1 and 5)

);

* Redémarrer Tomcat et visualiser le résultat de l’exécution sur le navigateur.

En résumé

Pour retrouver le code et les liens utiles de cette démonstration, copier le fichier demonstration1.html dans le répertoire /WebContent/modules/module4/demonstrations.