

Lenguajes Visuales

JAVA y JDBC

Ultima modificación el miércoles 3 de mayo de 2006 a las 16:00:53
Copyright © 2006, Kronoman – In loving memory of my father - <http://kronoman.kicks-ass.org/apuntes/>

Los drivers para diferentes DBMS los podemos obtener de
<http://developers.sun.com/product/jdbc/drivers/index.html>

Info sobre JDBC la ponemos obtener de <http://java.sun.com/products/jdbc/>

JDBC

Tutorial oficial

<http://java.sun.com/docs/books/tutorial/jdbc/basics/index.html>

Drivers

Donde obtener los drivers (los .jar, deben ser incluidos en el proyecto que hacemos en Eclipse)

- * MySQL : <http://www.mysql.com/products/connector/j/>
- * Microsoft SQL : <http://msdn.microsoft.com/data/jdbc/>

Programando

```
// Establecer una conexion con M$-SQL consta de los siguientes pasos:
// Notas: si en vez de MS SQL queremos usar MySQL
// el driver es Class.forName("com.mysql.jdbc.Driver");
// y el URL del server es jdbc:mysql://host:port/database
// ejemplo : jdbc:mysql://localhost/test
// mas info : http://dev.mysql.com/doc/refman/5.0/en/cj-configuration-properties.html

// Cargar el driver de MS SQL
Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");

// Establecer la conexion
// Aqui lo complicado es el URL, para MS SQL la forma es jdbc:sqlserver://serverName:port
// Ejemplo : jdbc:sqlserver://localhost
// mas info : http://msdn2.microsoft.com/en-us/library/ms378428.aspx
Connection con = DriverManager.getConnection(url, "myLogin", "myPassword");

// Haciendo una QUERY (ej: un SELECT * FROM blah) - esto es igual en MS SQL y MySQL
// ver el ejemplo de http://dev.mysql.com/doc/connector/j/en/cj-using-statements.html

Statement stmt = null;
ResultSet rs = null;

stmt = con.createStatement();
rs = stmt.executeQuery("SELECT * FROM blah");

// Y para mirar el contenido del ResultSet, un ejemplo :
while (rs.next())
{
    String s = rs.getString("CampoVarchar");
    float n = rs.getFloat("CampoFloat");
    System.out.println(s + " " + n);
}

// Si no sabemos la cantidad de campos del ResultSet, la metadata nos lo da
// pudiendo codear un visualizador generico de SELECTs
ResultSetMetaData rsmd = rs.getMetaData(); // datos del resultado (metadatos)

int columnas = rsmd.getColumnCount(); // cantidad de columnas
```

```

// poner encabezado
for (int i = 0; i < columnas; i++)
    System.out.println(rsmd.getColumnLabel(i+1) + " \t ");

System.out.println("\n\n");

// poner datos
while (r.next())
{
    for (int i = 0; i < columnas; i++)
        System.out.println(rs.getString(i+1) + " \t ");

    System.out.println("\n\n");
}

```

Clase Conexion.java de ejemplo

```

/*
 * Clase para conectarse a un DBMS
 * Por Kronoman
 * Mayo/2006
 * In loving memory of my father
 * Creado para la clase de Lenguajes Visuales de UADE
 *
 * Donde obtener los drivers (los .jar, deben ser incluidos en el proyecto que hacemos en Eclipse)
 * MySQL : http://www.mysql.com/products/connector/j/
 * Microsoft SQL : http://msdn.microsoft.com/data/jdbc/
 */

import java.sql.*;

public class Conexion
{
    private Connection con;

    public static final String DRIVER_MySQL = "com.mysql.jdbc.Driver"; // driver MySQL
    public static final String DRIVER_MsSQL = "com.microsoft.sqlserver.jdbc.SQLServerDriver"; // driver MSSQL

    public Conexion(String driver)
    {
        try {
            Class.forName(driver);
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }

    public void conectar(String url, String user, String pass) throws SQLException
    {
        // Aqui lo complicado es el URL, para MS SQL la forma es jdbc:sqlserver://serverName:port
        // Para MySQL es jdbc:mysql://host:port/database
        con = DriverManager.getConnection(url, user, pass);
    }

    public void desconectar() throws SQLException
    {
        con.close();
    }

    // ejecuta una Query y devuelve el resultado (ej : SELECT)
    public ResultSet ejecutarQuery(String sql) throws SQLException
    {
        Statement st = con.createStatement();
        return st.executeQuery(sql);
    }

    // ejecuta una sentencia, sin resultados (ej: UPDATE)
    public void ejecutar(String sql) throws SQLException
    {
        Statement st = con.createStatement();
        st.execute(sql);
    }

    // ejecuta una sentencia (puede o no devolver resultados)
    // devuelve null si no hay resultados, o un ResultSet
    public ResultSet ejecutarAmbos(String sql) throws SQLException
    {
        Statement st = con.createStatement();

        if (st.execute(sql))
            return st.getResultSet();

        return null;
    }
}

```