

```
1 // Class DataSourceBean.java
2 // Copyright 2001 ASR Strategic Resources
3 // AITP NCC 2001 Java Competition Event
4 // Java Contest Problem Statement
5 // Design: Don Baldwin & Craig Slinkman PhD
6 // Code: Eugene Wasserman & Don Baldwin
7
8 package jdbc;
9
10 import java.util.*;
11 import java.sql.*;
12 import java.io.Serializable;
13
14 public class DataSourceBean extends Object implements Serializable {
15     private static DataSourceBean dsb = null;
16     private boolean debug = false;
17     private Statement statement = null;
18     private Connection connection;
19     //private JToolBar bbar = new JToolBar();
20
21
22     private String url = "jdbc:odbc:contest";
23     private String driver = "sun.jdbc.odbc.JdbcOdbcDriver";
24     private String user = null;
25     private String pass = null;
26
27     private DataSourceBean() {
28
29     }
30
31     public static DataSourceBean getDataSourceBean() {
32         return getDataSourceBean(false);
33     }
34
35     public static DataSourceBean getDataSourceBean(boolean debug) {
36         if(null == dsb)
37             dsb = new DataSourceBean();
38         dsb.debug = debug;
39         return dsb;
40     }
41
42     /**
43     * returns the connection to the current database (using the current DataSourceInfo).
44     */
45     public Connection getConnection() {
46         try {
```

```
47         if(!isConnected()) {
48             //System.out.println("Not connected to database, will try to connect.");
49             connect();
50         }
51     } catch(Exception e) {e.printStackTrace();}
52     return connection;
53 }
54
55
56
57 /**
58  * connect to the Data Source described by the DataSourceDescriptor dsd
59  * should throw an exception when the connection fails
60  */
61 public void connect() throws SQLException, ClassNotFoundException,
62     InstantiationException, IllegalAccessException {
63     Connection con = null;
64     //Class.forName((String)iDSD.getDriver());
65
66     if(debug)
67         System.out.println("Connecting to url: " + url);
68     DriverManager.registerDriver(
69         (Driver)(Class.forName(driver)).newInstance());
70     if(debug)
71         System.out.println("Loaded driver: " + driver);
72     con = DriverManager.getConnection(url,
73         user, pass);
74     if(debug)
75         System.out.println("Made connection to: " + con);
76
77     this.connection = con;
78     this.statement = connection.createStatement();
79     if(debug)
80         System.out.println("Created statement: " + statement);
81
82 }
83
84 public void closeConnection() throws SQLException {
85     getConnection().close();
86 }
87
88 /**
89  * sets the current statement.
90  */
91 private void setStatement(Statement statement) {
92     this.statement = statement;
```

```
93     }
94
95     /**
96     * returns the current statement.
97     */
98     public Statement getStatement() {
99         getConnection();
100        return statement;
101    }
102
103
104
105     /**
106     * Execute an SQL query returning the result as a ResultSet, also throwing an exception
107     * if one is raised
108     */
109     public ResultSet executeQuery(String sql) throws SQLException {
110         if(debug)
111             System.out.println("executeQuery: " + sql);
112         getConnection();
113
114         ResultSet rs = getStatement().executeQuery(sql);
115         return rs;
116     }
117
118     /**
119     * Execute an SQL update returning the result as an int, also throwing an exception
120     * if one is raised
121     */
122     public int executeUpdate(String sql) throws SQLException {
123         if(debug)
124             System.out.println("executeUpdate: " + sql);
125         getConnection();
126         int lines = getStatement().executeUpdate(sql);
127         return lines;
128     }
129
130     /**
131     * returns wether there is a valid (instantiated) connection
132     */
133     public boolean isConnected() {
134         boolean ret = false;
135         try {
136             if(null == connection || connection.isClosed())
137                 ret = false;
138             else
```

```
139         ret = true;
140     } catch(SQLException e) {
141         e.printStackTrace();
142         ret = false;
143     }
144     return ret;
145 }
146
147 /**
148  * return the DatabaseMetaData for this connection
149  */
150 public DatabaseMetaData getMetaData() throws SQLException {
151     if(isConnected())
152         return getConnection().getMetaData();
153     else
154         return null;
155 }
156
157 /**
158  * return a String representation of the DataSource
159  */
160 public String toString() {
161     return "Connection to: " + url;
162 }
163
164 } // END Class DataSourceBean.java
```