[**Connect to Microsoft SQL Server via JDBC**](http://www.codejava.net/java-se/jdbc/connect-to-microsoft-sql-server-via-jdbc)

Last Updated on 12 January 2016   |   [Print](http://www.codejava.net/java-se/jdbc/connect-to-microsoft-sql-server-via-jdbc?tmpl=component&print=1&page=) [Email](http://www.codejava.net/component/mailto/?tmpl=component&template=protostar&link=ce77e92250f719c10260df749b9b2e142846f7e2)

|  |
| --- |
| [**Download Aspose for all your file format manipulation needs**](http://www.aspose.com/java/total-component.aspx?utm_source=CodeJava&utm_medium=banner&utm_campaign=CodeJava-Textforums) |

**Table of content:**

* 1. [Download Microsoft JDBC driver](http://www.codejava.net/java-se/jdbc/connect-to-microsoft-sql-server-via-jdbc#DownloadDriver)
  2. [JDBC database URL for SQL Server](http://www.codejava.net/java-se/jdbc/connect-to-microsoft-sql-server-via-jdbc#DatabaseURL)
  3. [Register JDBC driver for SQL Server and establish connection](http://www.codejava.net/java-se/jdbc/connect-to-microsoft-sql-server-via-jdbc#MakeConnection)
  4. [Example program](http://www.codejava.net/java-se/jdbc/connect-to-microsoft-sql-server-via-jdbc#ExampleProgram)

This article describes how to get JDBC driver and write code for making database connection to Microsoft SQL Server from a Java client. Suppose you have a light weight version of SQL Server installed, such as [Microsoft SQL Server 2012 Express](http://www.microsoft.com/en-us/download/details.aspx?id=29062).

**1.    Download Microsoft JDBC driver**

[Click here](http://www.microsoft.com/en-us/download/details.aspx?displaylang=en&id=11774) to download Microsoft JDBC Driver 4.0 for SQL Server which supports:

* + SQL Server versions: 2005, 2008, 2008 R2, and 2012.
  + JDK version: 5.0 and 6.0.

Run the downloaded program *sqljdbc\_<version>\_<language>.exe*. It will extract the files into a specified directory (default is*Microsoft JDBC Driver 4.0 for SQL Server*). You will find two jar files sqljdbc.jar (for JDBC 3.0) and sqljdbc4.jar (for JDBC 4.0), plus some .dll files and HTML help files.

Place the sqljdbc.jar file under your application’s classpath if you are using JDK 5.0 or sqljdbc4.jar file if you are using JDK 6.0 or later.

Recommended Book: [**Introducing Microsoft SQL Server 2012**](http://www.amazon.com/gp/product/073566515X/ref=as_li_tf_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=073566515X&linkCode=as2&tag=codejava-article-20)**http://ir-na.amazon-adsystem.com/e/ir?t=code0ac-20&l=as2&o=1&a=073566515X**

**2.    JDBC database URL for SQL Server**

The syntax of database URL for SQL Server is as follows:

***jdbc:sqlserver://****[serverName[\instanceName][:portNumber]][;property=value[;property=value]]*

Where:

* + serverName: host name or IP address of the machine on which SQL server is running.
  + instanceName: name of the instance to connect to on serverName. The default instance is used if this parameter is not specified.
  + portNumber: port number of SQL server, default is 1433. If this parameter is missing, the default port is used.
  + property=value: specify one or more additional connection properties. To see the properties specific to SQL server, visit [Setting the Connection Properties](http://msdn.microsoft.com/en-us/library/ms378988%28v=sql.90%29.aspx).

**NOTE:** SQL Server has two authentication modes:

* + Windows authentication: using current Windows user account to log on SQL Server. This mode is for the case both the client and the SQL server are running on the same machine. We specify this mode by adding the property*integratedSecurity=true* to the URL.
  + SQL Server authentication: using a SQL Server account to authenticate. We have to specify username and password explicitly for this mode.

Following are some examples:

-          Connect to default instance of SQL server running on the same machine as the JDBC client, using Windows authentication:

*jdbc:sqlserver://localhost;integratedSecurity=true;*

-          Connect to an instance named sqlexpress on the host dbServer, using SQL Server authentication:

*jdbc:sqlserver://dbHost\sqlexpress;user=sa;password=secret*

-          Connect to a named database testdb on localhost using Windows authentication:

*jdbc:sqlserver://localhost:1433;databaseName=testdb;integratedSecurity=true;*

Recommended Book: [**Microsoft SQL Server 2012 T-SQL Fundamentals**](http://www.amazon.com/gp/product/0735658145/ref=as_li_tf_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=0735658145&linkCode=as2&tag=codejava-article-20)**http://ir-na.amazon-adsystem.com/e/ir?t=code0ac-20&l=as2&o=1&a=0735658145**

**3.    Register JDBC driver for SQL Server and establish connection**

The JDBC driver class of SQL Server is com.microsoft.sqlserver.jdbc.SQLServerDriver, so to register this driver, use the following statement:

|  |  |
| --- | --- |
| 1 | DriverManager.registerDriver(new com.microsoft.sqlserver.jdbc.SQLServerDriver()); |

Or:

|  |  |
| --- | --- |
| 1 | Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver"); |

However, that is not required since JDBC 4.0 (JDK 6.0) because the driver manager can detect and load the driver class automatically as long as the sqljdbc4.jar is present in the classpath.

To make a connection, call the method getConnection() of the DriverManager class. Here is a code snippet that connects the user sa with password secret to the instance sqlexpress on localhost:

|  |  |
| --- | --- |
| 1  2  3  4  5 | String dbURL = "jdbc:sqlserver://localhost\\sqlexpress;user=sa;password=secret";  Connection conn = DriverManager.getConnection(dbURL);  if (conn != null) {      System.out.println("Connected");  } |

The following code passes username and password as arguments to the method getConnection():

|  |  |
| --- | --- |
| 1  2  3  4 | String dbURL = "jdbc:sqlserver://localhost\\sqlexpress";  String user = "sa";  String pass = "secret";  conn = DriverManager.getConnection(dbURL, user, pass); |

We can also use a java.util.Properties object to store connection properties, as in the following example:

|  |  |
| --- | --- |
| 1  2  3  4  5 | String dbURL = "jdbc:sqlserver://localhost\\sqlexpress";  Properties properties = new Properties();  properties.put("user", "sa");  properties.put("password", "secret");  conn = DriverManager.getConnection(dbURL, properties); |

**NOTE:** if you want to use Windows authentication mode (*integratedSecurity=true*), you must have the sqljdbc\_auth.dllin the classpath.

Recommended Book: [**Java Database Best Practices**](http://www.amazon.com/gp/product/0596005229/ref=as_li_tf_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=0596005229&linkCode=as2&tag=code0ac-20)**http://www.assoc-amazon.com/e/ir?t=code0ac-20&l=as2&o=1&a=0596005229**

**4.    Example program**

To demonstrate, we create a small program that connects to an SQL Server instance on localhost and print out some database information as follows:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46 | package net.codejava.jdbc;    import java.sql.Connection;  import java.sql.DatabaseMetaData;  import java.sql.DriverManager;  import java.sql.SQLException;    /\*\*   \* This program demonstrates how to establish database connection to Microsoft   \* SQL Server.   \* @author www.codejava.net   \*   \*/  public class JdbcSQLServerConnection {        public static void main(String[] args) {            Connection conn = null;            try {                String dbURL = "jdbc:sqlserver://localhost\\sqlexpress";              String user = "sa";              String pass = "secret";              conn = DriverManager.getConnection(dbURL, user, pass);              if (conn != null) {                  DatabaseMetaData dm = (DatabaseMetaData) conn.getMetaData();                  System.out.println("Driver name: " + dm.getDriverName());                  System.out.println("Driver version: " + dm.getDriverVersion());                  System.out.println("Product name: " + dm.getDatabaseProductName());                  System.out.println("Product version: " + dm.getDatabaseProductVersion());              }            } catch (SQLException ex) {              ex.printStackTrace();          } finally {              try {                  if (conn != null && !conn.isClosed()) {                      conn.close();                  }              } catch (SQLException ex) {                  ex.printStackTrace();              }          }      }  } |

The program would produce the following output:

Driver name: Microsoft JDBC Driver 4.0 for SQL Server

Driver version: 4.0.2206.100

Product name: Microsoft SQL Server

Product version: 11.00.2100