Retrieve data example

by Byron Kiourtzoglou on November 11th, 2012 | **Filed in:** sql **Tags:** core java, sql

This is an example of how to retrieve data from a database. Retrieving data from a database implies that you should:

- Load the JDBC driver, using the `forName(String className)` API method of theClass. In this example we use the MySQL JDBC driver.
- Create a Connection to the database. Invoke the `getConnection(String url, String user, String password)` API method of the DriverManager to create the connection.
- Create a Statement, using the `createStatement()` API method of the Connection.
- Execute the query to the database, using the `executeQuery(String sql)` API method. The data produced by the given query is a ResultSet.
- For each row of the result set, get the data of a column, using the `next()` and the `getString(String columnLabel)` API methods of the ResultSet. Note that ResultSet API provides appropriate methods for retrieving data according to the datatype, such as `getBoolean(String columnLabel)`, `getByte(String columnLabel)`, `getShort(String columnLabel)`, `getDouble(String columnLabel)`, `getDate(String columnLabel)`. We can also get the data from the current row using the column index with the `getString(int columnIndex)` API method.

Let's take a look at the code snippet that follows:

```
001 package com.javacodegeeks.snippets.core;
002
003 import java.sql.Connection;
004 import java.sql.DriverManager;
005 import java.sql.ResultSet;
006 import java.sql.SQLException;
007 import java.sql.Statement;
008
009 public class SelectRowsExample {
010
011   public static void main(String[] args) {
012
013     Connection connection = null;
014     try {
015
016   // Load the MySQL JDBC driver
017
018   String driverName = "com.mysql.jdbc.Driver";
019
020   Class.forName(driverName);
021
022
023   // Create a connection to the database
024
025   String serverName = "localhost";
026
027   String schema = "test";
028
029   String url = "jdbc:mysql://" + serverName +  "/" + schema;
030
031   String username = "username";
032
033   String password = "password";
034
035   connection = DriverManager.getConnection(url, username, password);
036
037
038
039   System.out.println("Successfully Connected to the database!");
040
041
042     } catch (ClassNotFoundException e) {
043
044   System.out.println("Could not find the database driver " + e.getMessage());
045     } catch (SQLException e) {
046
047   System.out.println("Could not connect to the database " + e.getMessage());
048     }
049
```

```
050     try {
051
052
053 // Get a result set containing all data from test_table
054
055 Statement statement = connection.createStatement();
056
057 ResultSet results = statement.executeQuery("SELECT * FROM test_table");
058
059
060 // For each row of the result set ...
061
062 while (results.next()) {
063
064
065   // Get the data from the current row using the column index - column data are in the VARCHAR format
066
067   String data = results.getString(1);
068
069   System.out.println("Fetching data by column index for row " + results.getRow() + " : " + data);
070
071
072   // Get the data from the current row using the column name - column data are in the VARCHAR format
073
074   data = results.getString("test_col");
075
076   System.out.println("Fetching data by column name for row " + results.getRow() + " : " + data);
077
078
079 }
080
081
082 /*
083
084   * Please note :
085
086   * ResultSet API provides appropriate methods for retrieving data
087
088   * based on each column data type e.g.
089
090   *
091
092   * boolean bool = rs.getBoolean("test_col");
093
094   * byte b = rs.getByte("test_col");
095
096   * short s = rs.getShort("test_col");
097
098   * int i = rs.getInt("test_col");
099
100   * long l = rs.getLong("test_col");
101
102   * float f = rs.getFloat("test_col");
103
104   * double d = rs.getDouble("test_col");
105
106   * BigDecimal bd = rs.getBigDecimal("test_col");
107
108   * String str = rs.getString("test_col");
109
110   * Date date = rs.getDate("test_col");
111
112   * Time t = rs.getTime("test_col");
113
114   * Timestamp ts = rs.getTimestamp("test_col");
115
116   * InputStream ais = rs.getAsciiStream("test_col");
117
118   * InputStream bis = rs.getBinaryStream("test_col");
119
120   * Blob blob = rs.getBlob("test_col");
121
122   */
123
124     } catch (SQLException e) {
```

```
125
126     System.out.println("Could not retrieve data from the database " + e.getMessage());
127       }
128
129     }
130 }
```

Example Output:

```
Successfully Connected to the database!
Fetching data by column index for row 1 : new_test_value
Fetching data by column name for row 1 : new_test_value
Fetching data by column index for row 2 : new_test_value_0
Fetching data by column name for row 2 : new_test_value_0
Fetching data by column index for row 3 : new_test_value_1
Fetching data by column name for row 3 : new_test_value_1
Fetching data by column index for row 4 : new_test_value_2
Fetching data by column name for row 4 : new_test_value_2
Fetching data by column index for row 5 : new_test_value_3
Fetching data by column name for row 5 : new_test_value_3
Fetching data by column index for row 6 : new_test_value_4
Fetching data by column name for row 6 : new_test_value_4
Fetching data by column index for row 7 : new_test_value_5
Fetching data by column name for row 7 : new_test_value_5
Fetching data by column index for row 8 : new_test_value_6
Fetching data by column name for row 8 : new_test_value_6
Fetching data by column index for row 9 : new_test_value_7
Fetching data by column name for row 9 : new_test_value_7
Fetching data by column index for row 10 : new_test_value_8
Fetching data by column name for row 10 : new_test_value_8
Fetching data by column index for row 11 : new_test_value_9
Fetching data by column name for row 11 : new_test_value_9
```

This was an example of how to retrieve data from a database in Java.