Calling MySQL Stored Procedures from JDBC

In this tutorial, you will learn how to call MySQL stored procedures from JDBC using CallableStatement object.

Before you start

For the sake of demonstration, we will create a new stored procedure named get\_candidate\_skill that accepts candidate\_id as the IN parameter and returns a result set that contains the skills of the candidate.

DELIMITER $$

CREATE PROCEDURE get\_candidate\_skill(IN candidate\_id INT)

BEGIN

SELECT candidates.id, first\_name,last\_name, skills.name AS skill

FROM candidates

INNER JOIN candidate\_skills ON candidates.id = candidate\_skills.candidate\_id

INNER JOIN skills ON skills.id = candidate\_skills.skill\_id

WHERE candidates.id = candidate\_id;

END$$

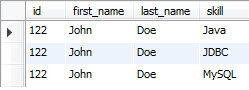
DELIMITER ;

Code language: SQL (Structured Query Language) (sql)

Let’s call this stored procedure for candidate id with value 122.

CALL get\_candidate\_skill(122);

Code language: SQL (Structured Query Language) (sql)



Introducing to CallableStatement and stored procedure call syntax

To call stored procedures or stored functions in MySQL from JDBC, you use CallableStatement object, which inherits from [PreparedStatement object](https://www.mysqltutorial.org/mysql-jdbc-update). The general syntax of calling a stored procedure is as follows:

{?= call procedure\_name(param1,param2,...)}

Code language: SQL (Structured Query Language) (sql)

You wrap the stored procedure call within braces ({}). If the stored procedure returns a value, you need to add the question mark and equal (?=) before the call keyword. If a stored procedure does not return any values, you just omit the ?= sign. In case the stored procedure accepts any parameters, you list them within the opening and closing parentheses after the stored procedure’s name.

The following are examples of using the syntax for calling stored procedures in different contexts:

|  |  |
| --- | --- |
| Syntax | Stores Procedures |
| {  call procedure\_name() } | Accept no parameters and return no value |
| { call procedure\_name(?,?) } | Accept two parameters and return no value |
| {?= call procedure\_name() } | Accept no parameter and return value |
| {?= call procedure\_name(?) } | Accept one parameter and return value |

Notice that question mark placeholder (?) can be used for both IN ,OUT, and INOUT parameters. For detailed information on different parameter types in stored procedures, check it out [MySQL stored procedure parameters tutorial](https://www.mysqltutorial.org/stored-procedures-parameters.aspx).

JDBC MySQL stored procedure example

First, open a connection to MySQL server by creating a new Connection object.

Connection conn = DriverManager.getConnection();

Code language: Java (java)

Then, prepare a stored procedure call and create a CallableStatement object by calling prepareCall() method of the Connection object.

String query = "{CALL get\_candidate\_skill(?)}";

CallableStatement stmt = conn.prepareCall(query)

Code language: Java (java)

Next, pass all the parameters to the stored procedure. In this case, the get\_candidate\_skill stored procedure accepts only one IN parameter.

stmt.setInt(1, candidateId);

Code language: Java (java)

After that, execute the stored procedure by calling the executeQuery() method of the CallableStatement object. It returns a result set in this case.

ResultSet rs = stmt.executeQuery();

Code language: Java (java)

Finally, traverse the ResultSet to display the results.

while (rs.next()) {

System.out.println(String.format("%s - %s",

rs.getString("first\_name") + " "

+ rs.getString("last\_name"),

rs.getString("skill")));

}

Code language: Java (java)

The following is the complete example of calling the MySQL stored procedure from JDBC.

package org.mysqltutorial;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.CallableStatement;

/\*\*

\*

\* @author mysqltutorial.org

\*/

public class Main {

/\*\*

\* Get skills by candidate id

\*

\* @param candidateId

\*/

public static void getSkills(int candidateId) {

//

String query = "{ call get\_candidate\_skill(?) }";

ResultSet rs;

try (Connection conn = MySQLJDBCUtil.getConnection();

CallableStatement stmt = conn.prepareCall(query)) {

stmt.setInt(1, candidateId);

rs = stmt.executeQuery();

while (rs.next()) {

System.out.println(String.format("%s - %s",

rs.getString("first\_name") + " "

+ rs.getString("last\_name"),

rs.getString("skill")));

}

} catch (SQLException ex) {

System.out.println(ex.getMessage());

}

}

/\*\*

\*

\* @param args

\*/

public static void main(String[] args) {

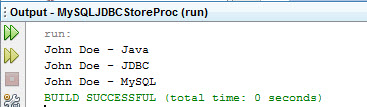
getSkills(122);

}

}

Code language: Java (java)

Let’s run the program.



The program works as expected.

In this tutorial, we have shown you how to call a stored procedure in MySQL database from a Java program using JDBC CallableStatement object.