

JSTL SQL <SQL:TRANSACTION> TAG

http://www.tutorialspoint.com/jsp/jstl_sql_transaction_tag.htm

Copyright © tutorialspoint.com

The <sql:transaction> tag is used to group <sql:query> and <sql:update> into transactions. You can put as many <sql:query> and <sql:update> as statements inside <sql:transaction> to make them a single transaction.

It ensures that the database modifications performed by the nested actions are either committed or rolled back if an exception is thrown by any nested action.

Attribute:

The <sql:transaction> tag has following attributes:

Attribute	Description	Required	Default
dataSource	Database connection to use <i>overrides the default</i>	No	Default database
isolation	Transaction isolation <i>READ_COMMITTED, READ_UNCOMMITTED, REPEATABLE_READ, or SERIALIZABLE</i>	No	Database's default

Example:

To start with basic concept, let us create a simple table **Students** table in TEST database and create few records in that table as follows:

Step 1:

Open a **Command Prompt** and change to the installation directory as follows:

```
C:\>
C:\>cd Program Files\MySQL\bin
C:\Program Files\MySQL\bin>
```

Step 2:

Login to database as follows

```
C:\Program Files\MySQL\bin>mysql -u root -p
Enter password: *****
mysql>
```

Step 3:

Create the table **Employee** in **TEST** database as follows:

```
mysql> use TEST;
mysql> create table Students
(
    id int not null,
    first varchar (255),
    last varchar (255),
    dob date
);
Query OK, 0 rows affected (0.08 sec)
mysql>
```

Create Data Records

Finally you create few records in Employee table as follows:

```
mysql> INSERT INTO Students
VALUES (100, 'Zara', 'Ali', '2002/05/16');
Query OK, 1 row affected (0.05 sec)

mysql> INSERT INTO Students
VALUES (101, 'Mahnaz', 'Fatma', '1978/11/28');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO Students
VALUES (102, 'Zaid', 'Khan', '1980/10/10');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Students
VALUES (103, 'Sumit', 'Mittal', '1971/05/08');
Query OK, 1 row affected (0.00 sec)

mysql>
```

Now let us write a JSP which will make use of <sql:update> along with <sql:transaction> to execute a SQL UPDATE statement. Here code inside <sql:transaction> either would be executed completely or not at all:

```
<%@ page import="java.io.*,java.util.*,java.sql.*"%>
<%@ page import="javax.servlet.http.*,javax.servlet.*"%>
<%@ page import="java.util.Date,java.text.*" %>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql"%>

<html>
<head>
<title>JSTL sql:transaction Tag</title>
</head>
<body>

<sql:setDataSource var="snapshot" driver="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost/TEST"
user="root" password="cohoneb"/>

<%
Date DoB = new Date("2001/12/16");
int studentId = 100;
%>

<sql:transaction dataSource="${snapshot}">
  <sql:update var="count">
    UPDATE Students SET last = 'Ali' WHERE Id = 102
  </sql:update>
  <sql:update var="count">
    UPDATE Students SET last = 'Shah' WHERE Id = 103
  </sql:update>
  <sql:update var="count">
    INSERT INTO Students
    VALUES (104, 'Nuha', 'Ali', '2010/05/26');
  </sql:update>
</sql:transaction>

<sql:query dataSource="${snapshot}" var="result">
  SELECT * from Students;
</sql:query>

<table border="1" width="100%">
<tr>
<th>Emp ID</th>
<th>First Name</th>
<th>Last Name</th>
<th>DoB</th>
</tr>
<c:forEach var="row" items="${result.rows}">
<tr>
<td><c:out value="${row.id}"/></td>
<td><c:out value="${row.first}"/></td>
<td><c:out value="${row.last}"/></td>
<td><c:out value="${row.dob}"/></td>
</tr>
</c:forEach>
</table>

</body>
</html>
```

Now try to access above JSP, which should display the following result:

Emp ID	First Name	Last Name	DoB
100	Zara	Ali	2001-12-16
101	Mahnaz	Fatma	1978-11-28
102	Zaid	Ali	1980-10-10
103	Sumit	Shah	1971-05-08
104	Nuha	Ali	2010-05-26

Loading [MathJax]/jax/output/HTML-CSS/jax.js