

Using Bean Shell Datalist Action to Duplicate & Delete Form Data

This post will demonstrate how to duplicate and delete form data onto another table through the use of bean shell datalist action.

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.sql.DataSource;
import org.joget.apps.app.service.AppUtil;
import org.joget.apps.form.model.FormRowSet;
import org.joget.commons.util.LogUtil;
import org.joget.commons.util.SecurityUtil;
import org.joget.apps.form.model.FormRow;
import org.joget.apps.form.model.FormRowSet;

FormRowSet rows = new FormRowSet();

// Loop through the selected rows and duplicate them to the destination datalist
for (String rowId : rowKeys ) {
    Connection con = null;
    try {
        // retrieve connection from the default datasource
        DataSource ds = (DataSource)AppUtil.getApplicationContext().getBean("setupDataSource");
        con = ds.getConnection();

        // execute SQL query
        if(!con.isClosed()) {

            FormRow row = new FormRow();
            //retrieve data for parent source
            String selectQuery = "SELECT c_name,c_options,c_date FROM app_fd_archive_requests WHERE id=?";
            PreparedStatement stmt = con.prepareStatement(selectQuery);
            stmt.setString(1, rowId);
            ResultSet rs = stmt.executeQuery();

            while (rs.next()) {
                row.setProperty("c_name", (rs.getObject("c_name") != null)?rs.getObject("c_name").
toString(): "");
                row.setProperty("c_options", (rs.getObject("c_options") != null)?rs.getObject("c_options").
toString(): "");
                row.setProperty("c_date", (rs.getObject("c_date") != null)?rs.getObject("c_date").
toString(): "");

                //store data for parent into archive
                String insertQuery = "INSERT INTO app_fd_archive_ar (id,c_name,c_options,c_date) values
(?,?,?,?)";

                PreparedStatement istmt = con.prepareStatement(insertQuery);
                istmt.setString(1,rowId);
                istmt.setString(2, row.getProperty("c_name"));
                istmt.setString(3, row.getProperty("c_options"));
                istmt.setString(4, row.getProperty("c_date"));
                istmt.executeUpdate();

                //break;
            }

            FormRow row2 = new FormRow();
            //retrieve data for child
            String selectChildQuery = "SELECT c_name,c_price,c_qty FROM app_fd_archive_requests_c WHERE
c_parent_id=?";

            PreparedStatement childstmt = con.prepareStatement(selectChildQuery);
            childstmt.setString(1, rowId);
            ResultSet rsChild = childstmt.executeQuery();
```

```

        while (rsChild.next()) {
            row2.setProperty("c_name", (rsChild.getObject("c_name") != null)?rsChild.getObject
("c_name").toString(): "");
            row2.setProperty("c_price", (rsChild.getObject("c_price") != null)?rsChild.getObject
("c_price").toString(): "");
            row2.setProperty("c_qty", (rsChild.getObject("c_qty") != null)?rsChild.getObject("c_qty").
toString(): "");

            //store data for child into child archive
            String insertChildQuery = "INSERT INTO app_fd_archive_ar_c (id,c_name,c_price,c_qty,
c_parent_id) values (UUID(),?,?,?,?)";
            PreparedStatement iChildstmt = con.prepareStatement(insertChildQuery);
            iChildstmt.setString(1, row2.getProperty("c_name"));
            iChildstmt.setString(2, row2.getProperty("c_price"));
            iChildstmt.setString(3, row2.getProperty("c_qty"));
            iChildstmt.setString(4, rowId);
            iChildstmt.executeUpdate();
            //break;
        }

        //delete parent source
        String deleteParentQuery = "DELETE FROM app_fd_archive_requests WHERE id=?";
        PreparedStatement deletestmt = con.prepareStatement(deleteParentQuery);
        deletestmt.setString(1, rowId);
        ResultSet rs = deletestmt.executeQuery();

        //delete child source
        String deleteChildQuery = "DELETE FROM app_fd_archive_requests_c WHERE c_parent_id=?";
        PreparedStatement dcstmt = con.prepareStatement(deleteChildQuery);
        dcstmt.setString(1, rowId);
        ResultSet rs = dcstmt.executeQuery();
    }
} catch (Exception e) {
    LogUtil.error("Archive app", e, "Error storing user data in bean shell datalist binder");
} finally {
    //always close the connection after used
    try {
        if (con != null) {
            con.close();
        }
    } catch (SQLException e) { /* ignored */ }
}
}

```

Form Design

Design of the form. Do keep in mind that this is made up of 2 forms (parent & child).

Section

Drag This Column

Name

Options

Date

Form Grid

Name

Quantity

Price

Userview

Before using Archive:

Admin Admin
imran.uzair@joget.com

Welcome

Manage Request

Manage Request

Manage Request - Child

Manage Archive Request

Manage Archive Request Child

Home > Manage Request > Manage Request

10

Show

<input type="checkbox"/>	NAME	OPTIONS	DATE	
<input type="checkbox"/>	Test	1	2023-05-01	Edit

One item found.

[CSV](#) | [Excel](#) | [XML](#) | [PDF](#)


Archive

New

Delete

Powered by Joget

After using Archive:

Admin Admin
imran.uzair@joget.com

Welcome

Manage Request

Manage Request0

Manage Request - Child0

Manage Archive Request1

Manage Archive Request Child1

Home > Manage Request > Manage Archive Request

10Show

<input type="checkbox"/>	NAME	OPTIONS	DATE	
<input type="checkbox"/>	Test	1	2023-05-01	Edit

One item found.

[CSV](#) | [Excel](#) | [XML](#) | [PDF](#)

NewDelete

Powered by Joget

Download the sample app below:

- [APP_kb_dx7_bean_shell_datalist_duplicate.jwa](#)