

Load and Store Form Grid Data Using Bean Shell Form Binder

This post explains how to load / read data from Form Grid using Bean Shell Form Binder. This can be handy if you need to load / store grid data from multiple tables.

Load Binder -> Bean Shell Form Binder

```
import org.joget.apps.form.model.*;
import org.joget.apps.form.service.*;
import org.joget.apps.app.service.*;
import java.sql.*;
import java.util.*;
import javax.sql.*;

FormRowSet f = new FormRowSet();
f.setMultiRow(true);

//Get Joget's current datasource configs
DataSource ds = (DataSource) AppUtil.getApplicationContext().getBean("setupDataSource");

con = ds.getConnection();

if (!con.isClosed()) {
    //Get the URL parameter
    String recordId = "#requestParam.id#";

    //Here you can query from one or multiple tables using JOIN etc
    String sql = "SELECT * FROM your_table_name WHERE id=?";
    PreparedStatement stmt = con.prepareStatement(sql);
    stmt.setString(1, recordId);

    //Execute the SELECT SQL statement
    ResultSet rs = stmt.executeQuery();

    //Get value from columns of record(s)
    while (rs.next()) {
        FormRow r1 = new FormRow();
        r1.put("gridColumn1", rs.getString(1));
        r1.put("gridColumn2", rs.getString(2));
        r1.put("gridColumn3", rs.getString(3));
        f.add(r1);
    }
}
con.close();
return f;
```

Store Binder -> Bean Shell Form Binder

```

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.Form;
import org.joget.apps.form.model.Element;
import org.joget.apps.form.model.FormData;
import org.joget.apps.form.model.FormRow;
import org.joget.apps.form.model.FormRowSet;
import org.joget.apps.form.service.FormUtil;
import org.joget.commons.util.UuidGenerator;

public saveGridRows(Element element, FormRowSet rows, FormData formData) {

    String recordId = null;
    Connection con = null;

    try {
        //Get Joget's current datasource configs
        DataSource ds = (DataSource) AppUtil.getApplicationContext().getBean("setupDataSource");

        con = ds.getConnection();

        if(!con.isClosed()) {
            //To generate new record IDs for storing into child table
            UuidGenerator uuid = UuidGenerator.getInstance();

            //Iterate to add new records
            Iterator i= rows.iterator();
            while (i.hasNext()) {
                FormRow row = (FormRow) i.next();

                String pId = uuid.getUuid();
                String gridColumn1 = row.get("gridColumn1");
                String gridColumn2 = row.get("gridColumn2");
                String gridColumn3 = row.get("gridColumn3");

                String insertSql = "INSERT INTO your_table_name (id,gridColumn1,gridColumn2,gridColumn3) VALUES
(?,?,?,?);";

                PreparedStatement stmtInsert = con.prepareStatement(insertSql);

                stmtInsert.setString(1, pId);
                stmtInsert.setString(2, gridColumn1);
                stmtInsert.setString(3, gridColumn2);
                stmtInsert.setString(4, gridColumn3);

                //Execute SQL statement
                stmtInsert.executeUpdate();
            }
        }
    } catch (Exception ex) {
        LogUtil.error("Your App/Plugin Name", ex, "Error storing using jdbc");
    } finally {
        try {
            if (con != null) {
                con.close();
            }
        } catch (Exception ex) {
            LogUtil.error("Your App/Plugin Name", ex, "Error closing the jdbc connection");
        }
    }
}

//Process and store grid rows
saveGridRows(element, rows, formData);

```

Grid Design

Correct Grid design should look like this:

Edit Grid
Edit Grid > UI & Validation > Data Binder > Load Binder (Bean Shell Form Binder)

ID *

gridId

Label

grid

Options

Value	Label	
gridColumn3	Whatever label	⬅ ➡ ✖
gridColumn1	Whatever label	⬅ ➡ ✖
gridColumn2	Whatever label	⬅ ➡ ✖

+

Example of a simple load binder beanshell script :

```
import org.joget.apps.form.model.FormRow;
import org.joget.apps.form.model.FormRowSet;

FormRowSet f = new FormRowSet();
f.setMultiRow(true);

FormRow r1 = new FormRow();
r1.put("gridColumn1", your_value);
r1.put("gridColumn2", your_value);
r1.put("gridColumn3", your_value);
f.add(r1);

FormRow r2 = new FormRow();
r2.put("gridColumn1", your_value);
r2.put("gridColumn2", your_value);
r2.put("gridColumn3", your_value);
f.add(r2);

return f;
```

Related Tutorial

- [Form Grid](#)
- [Bean Shell Form Binder](#)