

Load & 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 java.sql.*;
import org.apache.commons.collections.SequencedHashMap;
import java.util.*;

public FormRowSet test() {
    FormRowSet f = new FormRowSet();
    f.setMultiRow(true);
    Class.forName("com.mysql.jdbc.Driver").newInstance();
    con = DriverManager.getConnection("jdbc:mysql://localhost:3307/jwdb?characterEncoding=UTF-8", "database
username", "database password");

    if(!con.isClosed()){
        String recordId = "#requestParam.id#"; // Get the url parameter
        String sql = "SELECT * FROM test_table WHERE id=?"; // Here you can query from one or multiple
tables using JOIN etc
        PreparedStatement stmt = con.prepareStatement(sql);
        stmt.setString(1, recordId);
        ResultSet rs = stmt.executeQuery();
        while (rs.next()) {
            FormRow r1 = new FormRow();
            r1.put("gridColumn1", rs.getString(1));
            r1.put("gridColumn1", rs.getString(2));
            r1.put("gridColumn1", rs.getString(3));
            f.add(r1);
        }
    } else {
        System.out.println("Connection Problem");
    }
    con.close();
    return f;
}
return test();
```

Store Binder -> Bean Shell Form Binder

```

import org.joget.apps.form.model.*;
import org.joget.apps.form.lib.*;
import org.joget.apps.form.service.*;
import java.sql.*;
import org.apache.commons.collections.SequencedHashMap;
import java.util.*;
import org.joget.commons.util.UuidGenerator;

public FormRowSet getGridRows() {
    return rows ; // this will return the grid rows
}

public saveGridRows(FormRowSet rows) {
    Class.forName("com.mysql.jdbc.Driver").newInstance();
    con = DriverManager.getConnection("jdbc:mysql://localhost:3307/jwdb?characterEncoding=UTF-8", "database
username", "database password");
    if(!con.isClosed()){

        String recordId = "#requestParam.id#";

        // If you need to query data from different table(s) and process it before saving it to
database,
        // you can do it here

        UuidGenerator uuid = UuidGenerator.getInstance();

        Iterator i= rows.iterator(); // Iterating grid rows
        while (i.hasNext()) {
            FormRow row = (FormRow) i.next();
            String gridColumn1 = row.get("gridColumn1"); // reading grid column value
            String gridColumn2 = row.get("gridColumn2");
            String gridColumn3 = row.get("gridColumn3");
            String pId = uuid.getUuid(); // generating Primary Key
            String insertSql = "INSERT INTO table(id,col1,col2,col3) VALUES(?,?,?,?)";
            PreparedStatement stmtInsert = con.prepareStatement(insertSql);
            stmtInsert.setString(1, pId);
            stmtInsert.setString(2, gridColumn1);
            stmtInsert.setString(3, gridColumn2);
            stmtInsert.setString(4, gridColumn3);
            stmtInsert.executeUpdate();

        }
    } else {
        System.out.println("Connection Problem");
    }
    con.close();
}

FormRowSet rows = getGridRows(); // getting the grid rows
saveGridRows(rows); // processing & storing the grid rows

```

Grid Design

Correct Grid design should look like this:

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

ID *

Label

Options

Value	Label
<input type="text" value="gridColumn3"/>	<input type="text" value="Whatever label"/>
<input type="text" value="gridColumn1"/>	<input type="text" value="Whatever label"/>
<input type="text" value="gridColumn2"/>	<input type="text" value="Whatever label"/>

+

part of load binder beanshell script :

```
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);
```