Store Form Field Data to Multiple Tables

You may want to store certain fields from your form to other tables with the use of the Beanshell Form Binder. **Figure 1** shows an example of a form where the first 3 fields are to be stored in another data source in addition to the original form data table.

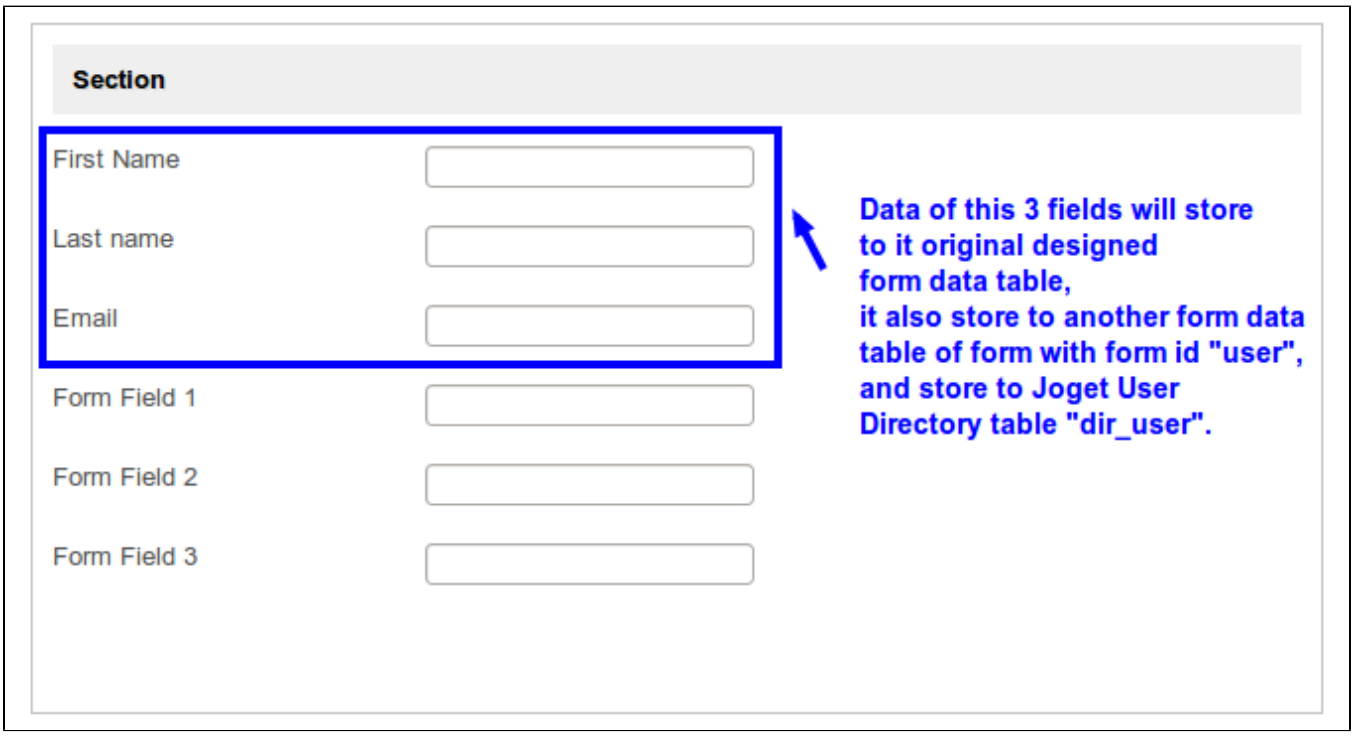

Figure1: Form with Field to Store

Then, click on the form "Properties" tab and navigates to "Advanced" page. Choose "Bean Shell Form Binder" as Store Binder.

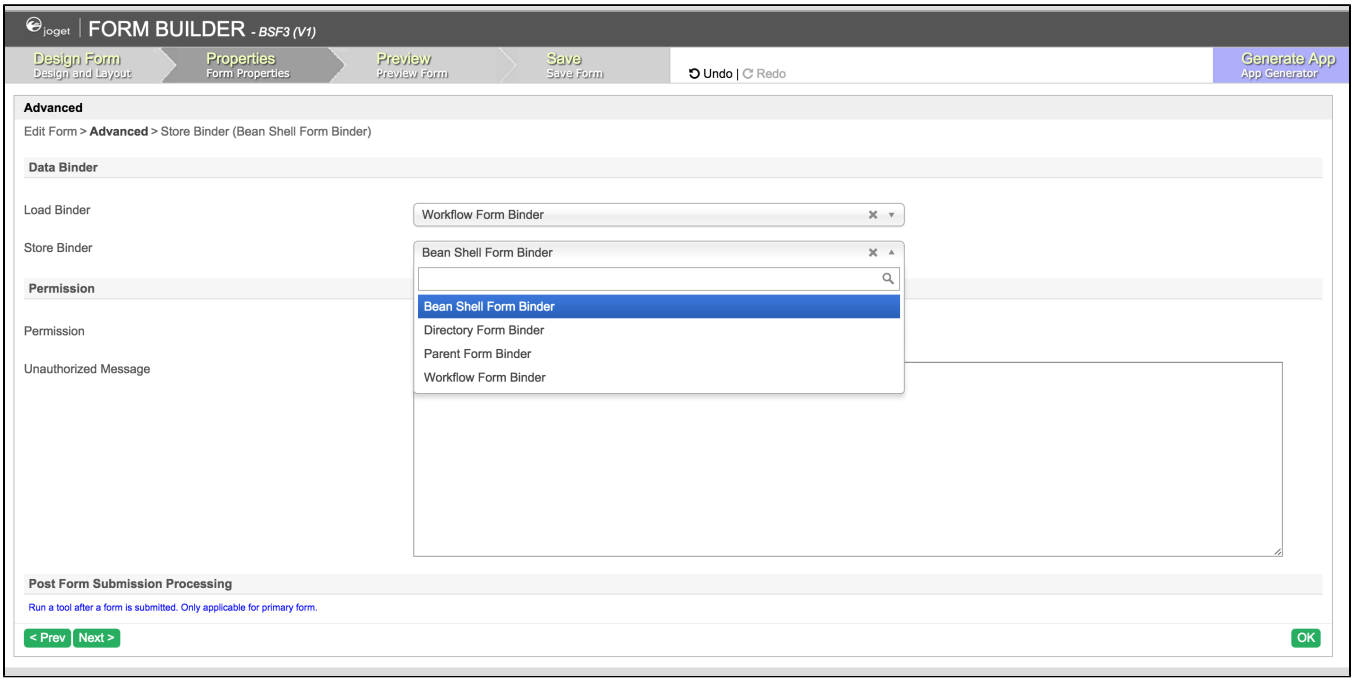

Figure 2: Choose Bean Shell Form Binder as the Store Binder

Configure Bean Shell Form Binder with your own coding to store the fields as intended, as shown in the figure below.

Code used in this example:

```
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.model.AppDefinition;
import org.joget.apps.app.service.AppService;
import org.joget.apps.app.service.AppUtil;
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.model.FormStoreBinder;
import org.joget.apps.form.service.FormUtil;
import org.joget.plugin.base.PluginManager;
import org.joget.commons.util.LogUtil;

public FormRowSet storeData(Element element, FormRowSet rows, FormData formData) {
    //check for empty data
    if (rows == null || rows.isEmpty()) {
        return rows;
    }
    normalStoring(element, rows, formData);

    //store only needed field by create new Form Row Set
    FormRow originalRow = rows.get(0);
    FormRowSet newRows = new FormRowSet();
    FormRow newRow = new FormRow();

    newRow.put("firstName", originalRow.getProperty("firstName"));
    newRow.put("lastName", originalRow.getProperty("lastName"));
    newRow.put("email", originalRow.getProperty("email"));
    newRows.add(newRow);

    String id = "#currentUser.username#";

    //store
    storeToOtherFormDataTable(element, newRows, formData, id);
    storeUsingJDBC(element, newRows, formData, id);

    return rows;
}

//this function will reuse workflow form binder to store data
public void normalStoring(Element element, FormRowSet rows, FormData formData) {
    PluginManager pluginManager = (PluginManager) AppUtil.getApplicationContext().getBean("pluginManager");
    FormStoreBinder binder = (FormStoreBinder) pluginManager.getPlugin("org.joget.apps.form.lib.
WorkflowFormBinder");
    binder.store(element, rows, formData);
}

//this function will store rows data to a form's data table
public void storeToOtherFormDataTable(Element element, FormRowSet rows, FormData formData, String id) {
    AppService appService = (AppService) AppUtil.getApplicationContext().getBean("appService");

    String formId = "user"; // the table of database is configured in the form with id "user"
    AppDefinition appDef = AppUtil.getCurrentAppDefinition();

    appService.storeFormData(appDef.getId(), appDef.getVersion().toString(), formId, rows, id);
}

//this function will store rows data to external source using JDBC
public void storeUsingJDBC(Element element, FormRowSet rows, FormData formData, String id) {
    Connection con = null;
    try {
        // retrieve connection from the default datasource
        DataSource ds = (DataSource)AppUtil.getApplicationContext().getBean("setupDataSource");
        con = ds.getConnection();

        if (!con.isClosed()) {
            //manually handle insert and update by checking the data is exist or not
```

```
            String selectQuery = "SELECT username FROM dir_user WHERE username=?";
            PreparedStatement stmt = con.prepareStatement(selectQuery);
            stmt.setString(1, id);
            ResultSet rs = stmt.executeQuery();

            Boolean isExist = false;
            if (rs.next()) {
                isExist = true;
            }

            FormRow row = rows.get(0);

            if (isExist) {
                String updateQuery = "UPDATE dir_user SET firstName = ?, lastName = ?, email = ? WHERE username
= ?";
                PreparedStatement ustmt = con.prepareStatement(updateQuery);
                ustmt.setString(1, row.getProperty("firstName"));
                ustmt.setString(2, row.getProperty("lastName"));
                ustmt.setString(3, row.getProperty("email"));
                ustmt.setString(4, id);
                ustmt.executeUpdate();
            } else {
                String insertQuery = "INSERT INTO dir_user (id, username, firstName, lastName, password, email)
values (?, ?, ?, ?, 'md5(password)', ?)";
                PreparedStatement istmt = con.prepareStatement(insertQuery);
                istmt.setString(1, id);
                istmt.setString(2, id);
                istmt.setString(3, row.getProperty("firstName"));
                istmt.setString(4, row.getProperty("lastName"));
                istmt.setString(5, row.getProperty("email"));
                istmt.executeUpdate();
            }
        }
    } catch (Exception e) {
        LogUtil.error("Sample app - StoreToMultipleSource form", e, "Error storing using jdbc");
    } finally {
        try {
            if(con != null) {
                con.close();
                        }
        } catch (SQLException e) {}
    }
}

//call storeData method with injected variables
return storeData(element, rows, formData);
```

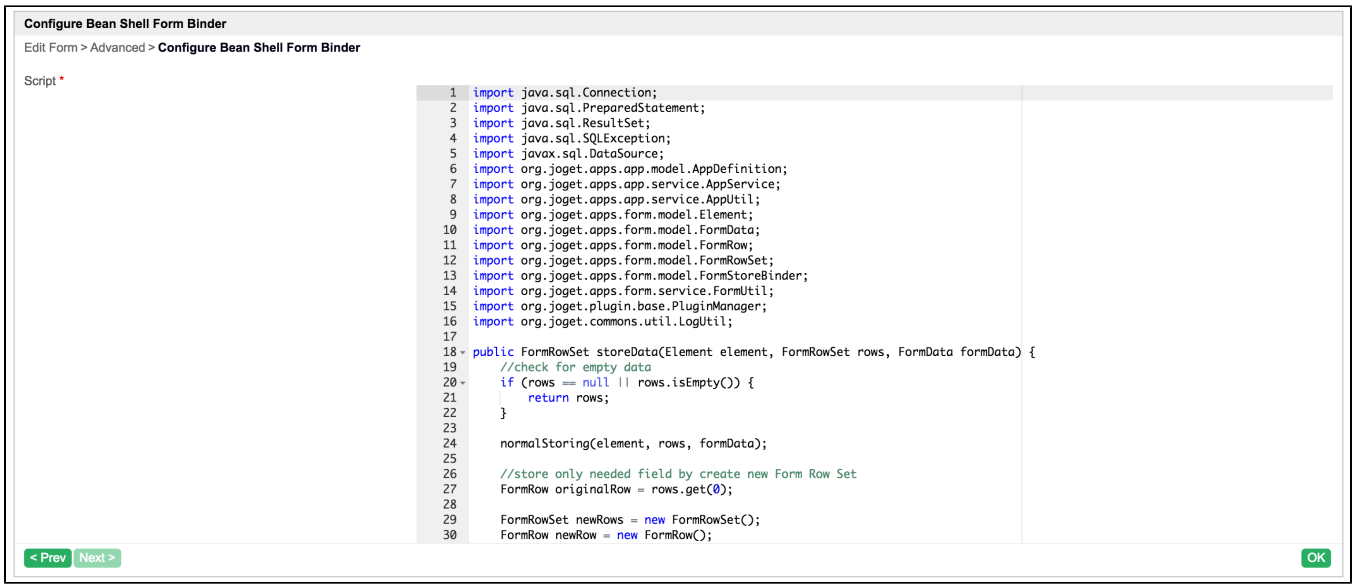

Figure 3: Populate Bean Shell Form Binder with the Necessary Codes

If the coding is properly written and tested, you will get this result:

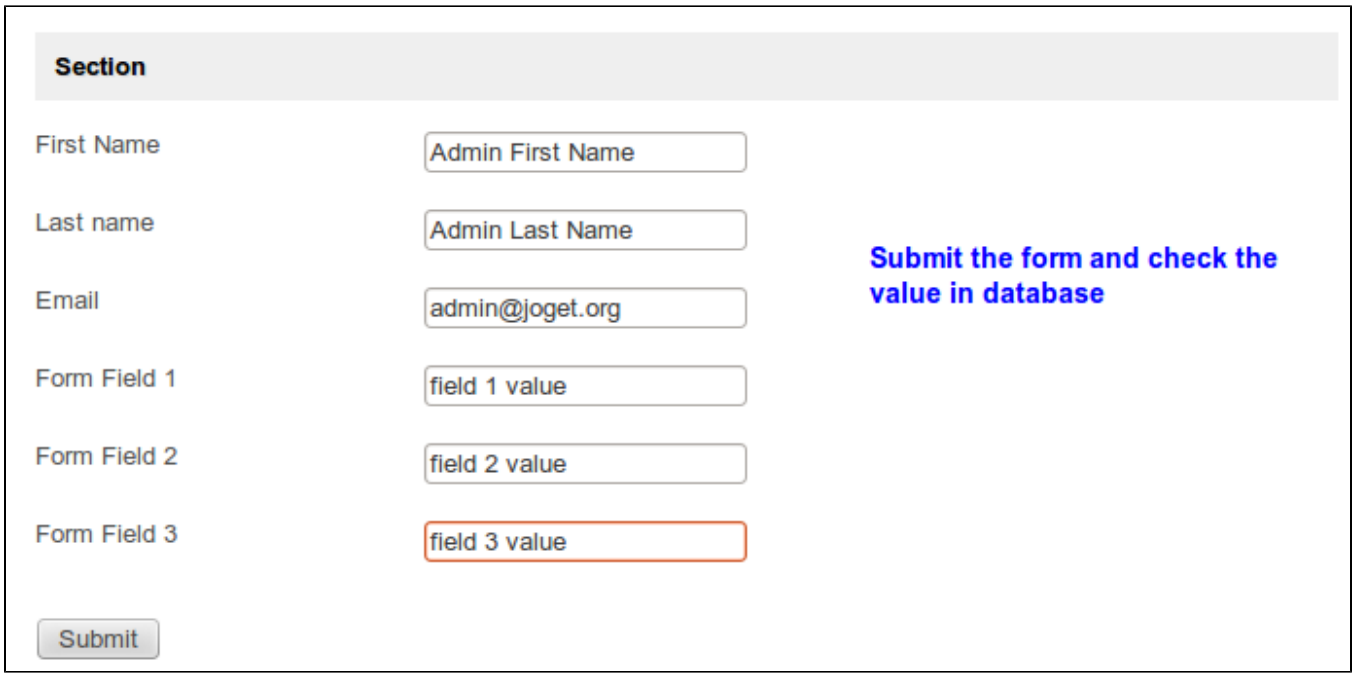

Figure 4: Fill and Submit Form for Testing

Check the data in the database.

```
mysql> select * from app_fd_customer \G
*************************** 1. row ***************************
          id: 2ce5be05-7f001010-ccf5ae80-e6d32b25
 dateCreated: 2012-06-27 15:44:11
dateModified: 2012-06-27 15:44:11
  c_lastName: Admin Last Name
    c_field3: field 3 value
    c_field2: field 2 value
     c_email: admin@joget.org
    c_field1: field 1 value
 c_firstName: Admin First Name
1 row in set (0.01 sec)

mysql> select * from app_fd_user \G
*************************** 1. row ***************************
          id: admin
 dateCreated: 2012-06-27 15:44:10
dateModified: 2012-06-27 15:44:10
  c_lastName: Admin Last Name
     c_email: admin@joget.org
 c_firstName: Admin First Name
1 row in set (0.00 sec)

mysql> select * from dir_user where username = 'admin' \G
*************************** 1. row ***************************
       id: admin
 username: admin
 password: 21232f297a57a5a743894a0e4a801fc3
firstName: Admin First Name
 lastName: Admin Last Name
    email: admin@joget.org
   active: 1
 timeZone: 0
1 row in set (0.00 sec)

mysql>
```

Figure 5: Data Stored Correctly in the Tables