

## Pre-populate Form Fields with Data from External Source

You may initialize a new form with some fields pre-populated using Bean Shell Form Binder.

Figure 1 below shows an example of a form whose first 3 fields are to be pre-populated.

The form interface includes the following elements:

- Section**: A header for the group of fields.
- First Name**: An input field.
- Last name**: An input field.
- Email**: An input field.
- Form Field 1**: An input field.
- Form Field 2**: An input field.
- Form Field 3**: An input field.

A blue box highlights the first three input fields (First Name, Last name, Email). To the right of this box, a blue callout text reads: **Prepopulate data for this 3 fields from External Source (eg: dir\_user)**.

Figure 1: Form with Fields to Pre-populate

The quick and easy approach in addressing this requirement is to make use of the Beanshell Form Binder in the section's Load Binder. Edit the section.

**Section**

*Drag This Column*

First Name	<input type="text"/>	<b>Edit Section</b>
Last Name	<input type="text"/>	
Email	<input type="text"/>	
Form Field 1	<input type="text"/>	
Form Field 2	<input type="text"/>	
Form Field 3	<input type="text"/>	

(+) 

Figure 2: Configuring Section Properties to Determine How Data Will Be Handled

In **Load Binder**, select "Bean Shell Form Binder" as the Load Binder.

**Data Binder**

Edit Section > Data Binder > Load Binder (Bean Shell Form Binder) > Advanced Options

Load Binder	Bean Shell Form Binder	X ▾
Store Binder		▼

Figure 3: Choose Beanshell Form Binder as the Load Binder

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

Code used in this example:

```

<%
    String first = "John";
    String last = "Doe";
    String email = "johndoe@example.com";
    String field1 = "Field 1";
    String field2 = "Field 2";
    String field3 = "Field 3";
%>

```

```

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

public FormRowSet getData() {

    //-----
    //In this part of code, it trying to load the original data from form data table.

    FormRowSet results = null;
    if (primaryKey != null && primaryKey.trim().length() > 0) {
        AppService appService = (AppService) FormUtil.getApplicationContext().getBean("appService");
        Form form = FormUtil.findRootForm(element);
        if (form.equals(element) && form.getParent() != null) {
            form = FormUtil.findRootForm(form.getParent());
        }
        if (form != null) {
            results = appService.loadFormDataWithoutTransaction(form, primaryKey);
        }
    }
    //-----

    //-----
    //In this second part of code, it will load the data from external source by using
    //JDBC. It will run only when the first part of code fail to retrieve data from
    //form data table. This example use dir_user table of Joget as external source.

    if (results == null) {
        results = new FormRowSet();

        Connection con = null;
        try {
            Class.forName("com.mysql.jdbc.Driver").newInstance();
            con = DriverManager.getConnection("jdbc:mysql://localhost:3306/jwdb2?characterEncoding=UTF-8",
"root", "root");

            if(!con.isClosed()){
                String pId = "#currentUser.username#";
                String sql = "SELECT firstName, lastName, email FROM dir_user WHERE username=?";
                PreparedStatement stmt = con.prepareStatement(sql);
                stmt.setString(1, pId);

                ResultSet rs = stmt.executeQuery();
                while (rs.next()) {
                    FormRow row = new FormRow();
                    row.put("firstName", (rs.getString(1) != null)?rs.getString(1):@"" );
                    row.put("lastName", (rs.getString(2) != null)?rs.getString(2):@"" );
                    row.put("email", (rs.getString(3) != null)?rs.getString(3):@"" );
                    results.add(row);
                }
            }
        } catch(Exception ex) {
            System.err.println("Exception: " + ex.getMessage());
        } finally {
            try {
                if(con != null)
                    con.close();
            } catch(SQLException e) {}
        }
    }
    //-----

    return results;
}

return getData();
}

```

The screenshot shows a dialog box titled "Configure Bean Shell Form Binder". At the top right, there is a "Save" button with the text "Auto save when close?". Below the title, the path "Edit Section > Data Binder > Configure Bean Shell Form Binder > Advanced Options" is displayed. The main area is a code editor titled "Script \*". The code is a Java script with line numbers from 1 to 29. It imports various org.joget.\* packages and java.sql.\*. The script contains two main sections: one for loading data from a form data table and another for loading data from an external source using JDBC. At the bottom of the dialog are buttons for "< Prev" and "Next >" on the left, and "OK" and "Cancel" on the right.

```
1 import org.joget.apps.app.service.*;
2 import org.joget.apps.form.model.*;
3 import org.joget.apps.form.service.*;
4 import java.sql.*;
5 import java.util.*;
6
7 public FormRowSet getData() {
8
9     //-----
10    //In this part of code, it trying to load the original data from form data tabl
11
12    FormRowSet results = null;
13    if (primaryKey != null && primaryKey.trim().length() > 0) {
14        AppService appService = (AppService) FormUtil.getApplicationContext().getBe
15        Form form = FormUtil.findRootForm(element);
16        if (form.equals(element) && form.getParent() != null) {
17            form = FormUtil.findRootForm(form.getParent());
18        }
19        if (form != null) {
20            results = appService.loadFormDataWithoutTransaction(form, primaryKey);
21        }
22    }
23    //-----
24
25    //-----
26    //In this second part of code, it will load the data from external source by us
27    ////JDBC. It will run only when the first part of code fail to retrieve data from
28    ////form data table. This example use dir_user table of Joget as external source.
29}
```

Figure 4: Populate Beanshell Form Binder with the Necessary Codes

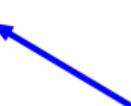
If the coding is properly written and tested, you should get this result:

**Workflow Process 1**

**Section**

First Name	Admin
Last name	Admin
Email	admin@joget.org
Form Field 1	
Form Field 2	
Form Field 3	

**Submit**



Data prepopulated in the field

Figure 5: The 3 Pre-populated Fields After a New Form is Loaded

#### Related Elements

- Bean Shell Form Binder