

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.

Section

First Name

Last name

Email

Form Field 1

Form Field 2

Form Field 3

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

Add Section

Add Column

Edit Section

Delete Section

Drag This Column

First Name

Last name

Email

Form Field 1

Form Field 2

Form Field 3

Edit Section

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

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

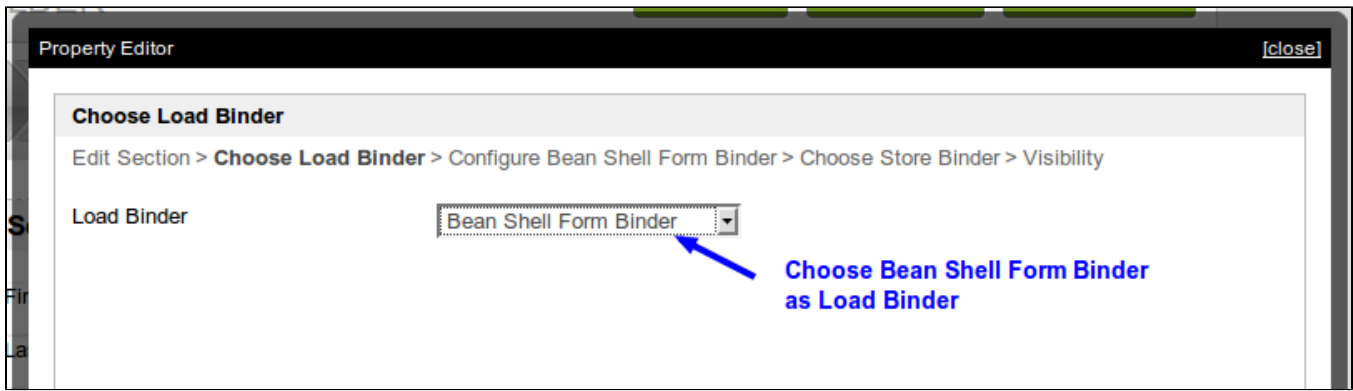


Figure 3: Choose Beanshell Form Binder as the Load Binder

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:



```

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();

```

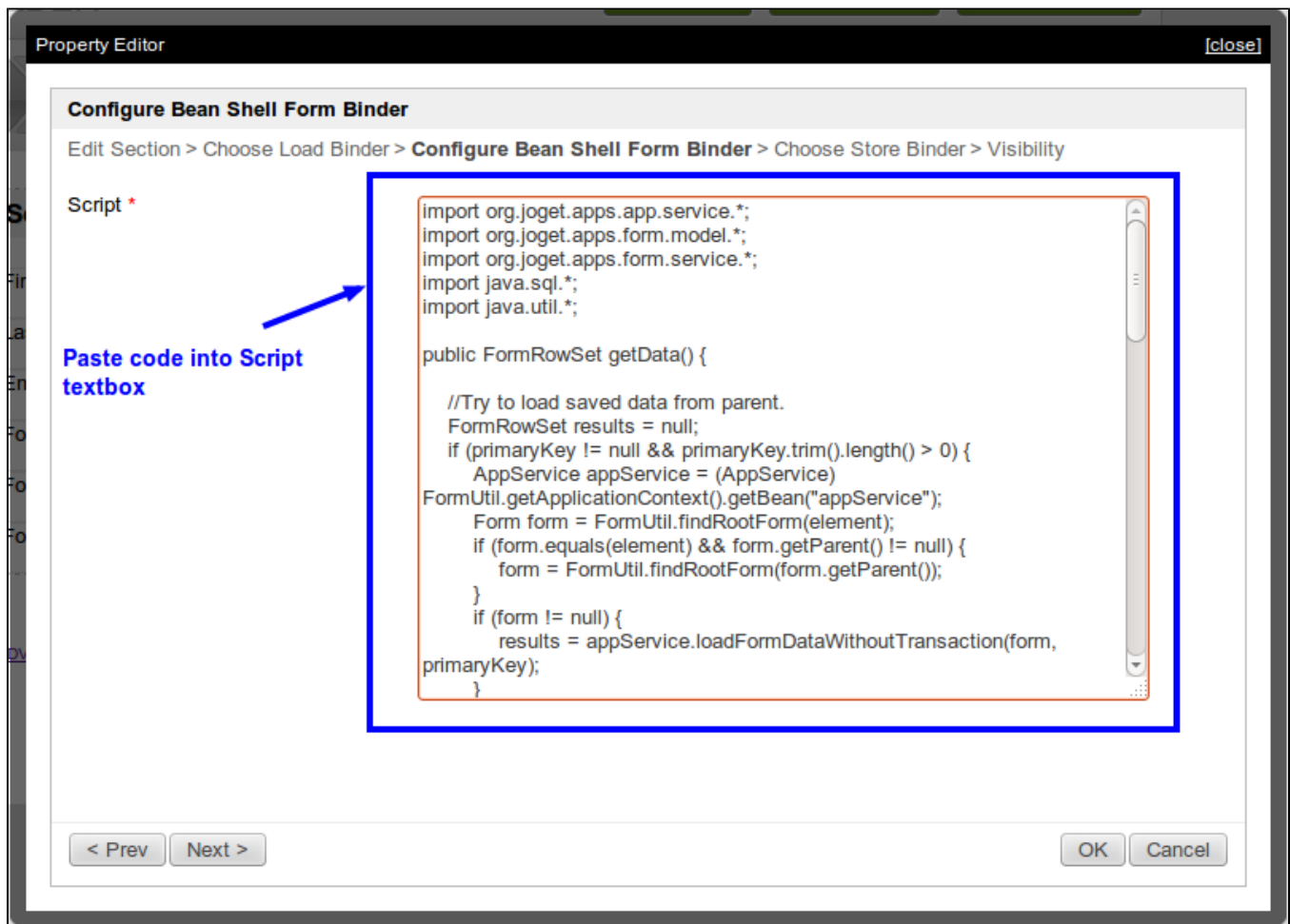


Figure 4: Populate Beanshell Form Binder with the Necessary Codes

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](#)