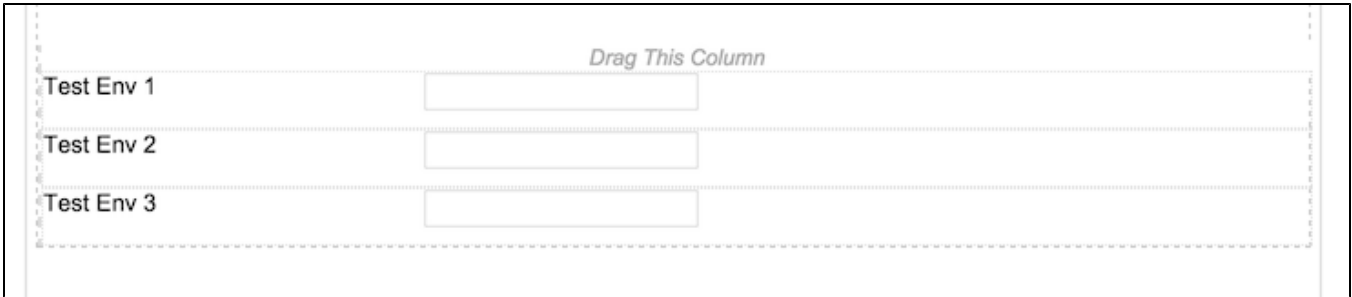


Manage Environment Variable using Form

By using Bean Shell Form Binder, we are able to load and update our [Environment Variable](#) values by using a form.

Let's create a form with each Text Field to represent an Environment Variable in our app and set all the field id to the ID of the Environment Variable.



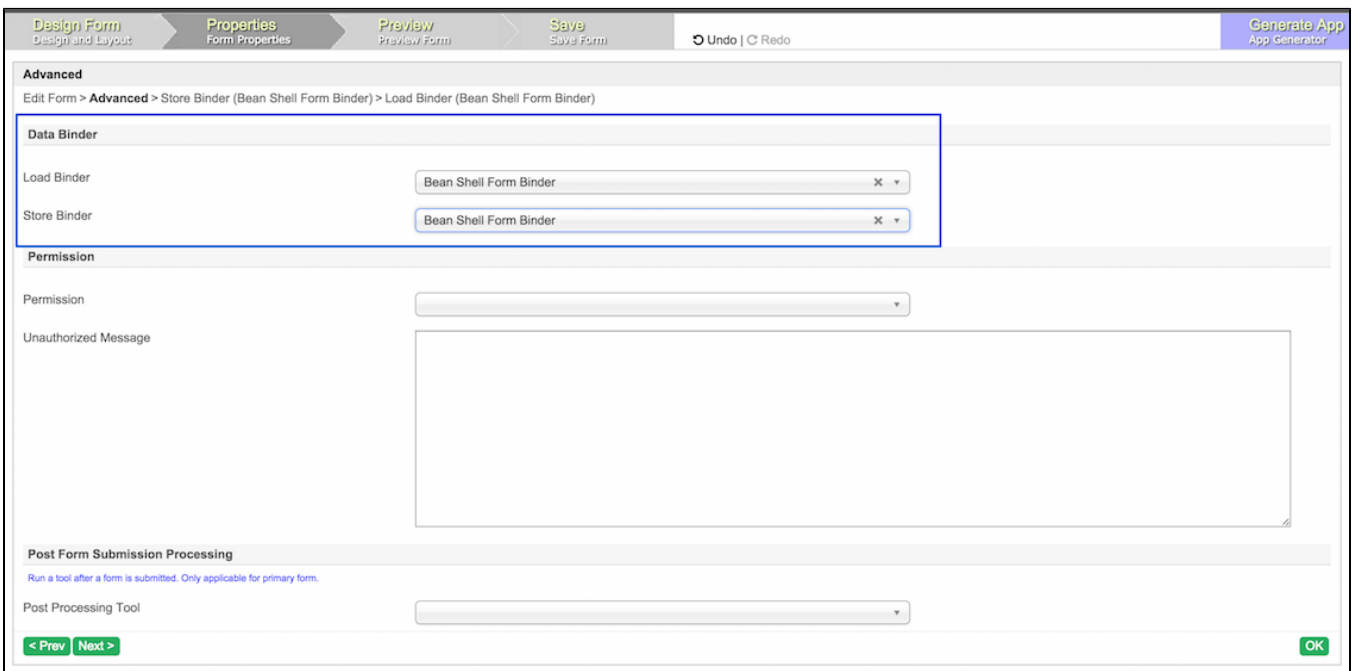
Test Env 1

Test Env 2

Test Env 3

Drag This Column

Then, click on "Properties" tab then set the "Load Binder" and "Store Binder" in "Advanced" page to "Bean Shell Form Binder".



Design Form Properties Preview Save Undo | C Redo Generate App App Generator

Advanced

Edit Form > Advanced > Store Binder (Bean Shell Form Binder) > Load Binder (Bean Shell Form Binder)

Data Binder

Load Binder Bean Shell Form Binder x v

Store Binder Bean Shell Form Binder x v

Permission

Permission

Unauthorized Message

Post Form Submission Processing

Run a tool after a form is submitted. Only applicable for primary form.

Post Processing Tool

< Prev Next > OK



Be advised the following BeanShell codes are **not compatible** in Joget Cloud multi-tenant environment.

Using the following script for "Load Binder".

```

import java.util.Collection;
import org.joget.apps.app.dao.EnvironmentVariableDao;
import org.joget.apps.app.model.AppDefinition;
import org.joget.apps.app.model.EnvironmentVariable;
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.commons.util.LogUtil;

public FormRowSet load(Element element, String primaryKey, FormData formData) {
    FormRowSet rows = new FormRowSet();
    AppDefinition appDef = AppUtil.getCurrentAppDefinition();

    if (appDef != null) {
        EnvironmentVariableDao environmentVariableDao = (EnvironmentVariableDao) AppUtil.
getApplicationContext().getBean("environmentVariableDao");
        Collection environmentVariableList = environmentVariableDao.getEnvironmentVariableList(null, appDef,
null, null, null, null);

        //loop the result and add to FormRow
        if (environmentVariableList != null && environmentVariableList.size() > 0) {
            FormRow r = new FormRow();

            for (EnvironmentVariable e : environmentVariableList) {
                if (e.getValue() != null) {
                    r.setProperty(e.getId(), e.getValue());
                }
            }

            rows.add(r);
        }
    }

    return rows;
}

//call load method with injected variable
return load(element, primaryKey, formData);

```

Using the following script for "Store Binder".

```

import org.joget.apps.app.dao.EnvironmentVariableDao;
import org.joget.apps.app.model.AppDefinition;
import org.joget.apps.app.model.EnvironmentVariable;
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.plugin.base.PluginManager;

public FormRowSet store(Element element, FormRowSet rows, FormData formData) {
    //check the rows is not empty before store it
    if (rows != null && !rows.isEmpty()) {
        AppDefinition appDef = AppUtil.getCurrentAppDefinition();
        EnvironmentVariableDao environmentVariableDao = (EnvironmentVariableDao) AppUtil.
getApplicationContext().getBean("environmentVariableDao");

        //Get the submitted data
        FormRow row = rows.get(0);

        //Add / Update each Environment Variable
        for (Object key : row.keySet()) {
            String envId = (String) key;
            String value = row.getProperty(envId);

            if (!envId.isEmpty()){
                EnvironmentVariable e = environmentVariableDao.loadById(envId, appDef);
                if (e != null) {
                    e.setValue(value);
                    environmentVariableDao.update(e);
                } else {
                    EnvironmentVariable e = new EnvironmentVariable();
                    e.setAppDefinition(appDef);
                    e.setId(envId);
                    e.setValue(value);
                    environmentVariableDao.add(e);
                }
            }
        }
    }

    return rows;
}

//call store method with injected variable
return store(element, rows, formData);

```

The resulted form as following.

The screenshot displays a web application interface for updating environment variables. The main heading is "Update Environment Variables by Form". The page is part of a "Setup" section, as indicated by the breadcrumb "Home > Setup". The user "Admin Admin" is logged in, shown in the top right corner. The form contains three input fields, each labeled with an environment variable name: "Test Env 1", "Test Env 2", and "Test Env 3". Each field contains the text "Test 1", "Test 2", and "Test 3" respectively. A blue "Save" button is positioned below the input fields. In the bottom right corner, there is a "Print" button with a printer icon.

Sample app for this article can download from [APP_update_env-1-20151027104612.jwa](#).