

Manage Environment Variable using Form

By using Bean Shell Form Data Store, 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 a 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

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

Advanced

Data

Load Data From

BeanShell

X

Save Data To


BeanShell

X

Permission

Permission

Unauthorized Message



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

Using the following script for "Load Data Store".

```

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 Data Store".

```

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 shows a web application titled "UPDATE ENVIRONMENT VARIABLES BY FORM". The interface includes a sidebar with a user profile "Admin Admin" and a "Setup" link. The main content area contains a form with three rows, each with a label and an input field:

Label	Input Field
Test Env 1	Test 1
Test Env 2	Test 2
Test Env 3	Test 3

At the bottom of the form is a blue "Save" button. In the bottom right corner of the page, there is a "Print" link.

Sample app for this article can download from [APP_update_env-1-20220901160243.jwa](#)

