

Using the Mura FW/1 Connector Plugin

Posted At : February 19, 2010 3:45 PM | Posted By : Bob Silverberg

Related Categories: FW/1, ColdFusion, Mura CMS

Several months ago [Pat Santora](#) of [Blue River](#) and I developed a Mura plugin that would allow a developer to take an existing **FW/1** application and deploy it within a Mura page. I was interested in this as I saw a need for a lightweight framework that I could use for my plugins, and by coincidence Pat happened to be working on something similar. So we joined forces and the **FW/1 Connector Plugin** was born.

The plugin was finally released yesterday so I thought it prudent to author a post about how to use the plugin.

Overview

The FW/1 Connector plugin can be used to incorporate an FW/1 application into Mura. Each plugin can be assigned to an individual FW/1 application. As long as each FW/1 application has a unique applicationKey, you can have as many FW/1 applications running inside Mura as you please.

Requirements

1. The **FW/1 framework.cfc** must be installed in a path that maps to *org.corfield*. You can do this via a mapping in the CF administrator or you can place the *org.corfield* folder in Mura's *requirements* folder.
2. Your FW/1 application must also reside in a path mapped to CF. Again, this can be accomplished via a mapping in the CF administrator or you can create a folder for your application in Mura's *requirements* folder.

Installing the Plugin

Download the latest version of the plugin from the [Mura App Store](#), and install it via the Mura Administrator.

From the Mura admin screen, click on *Site Settings*, on the right-hand side of the yellow menu bar. You should see two tabs, *Current Sites* and *Plugins*. Click the *Plugins* tab and you will see a list of any installed plugins, above which is a form that allows you to browse for a zip file on your local machine. Choose a the file that you downloaded and click the *Deploy* button. You'll see the Plugin Settings page for the FW/1 Connector Plugin. Provide values for the following settings:

1. **Plugin Name (Alias)** - This is the name that you're going to give to this instance of the plugin. Remember that you can install multiple instances of the plugin each of which can point to a different FW/1 application.
2. **FW/1 Application Key** - This must be the same value as specified in *variables.framework.applicationKey* inside your FW/1 application.
3. **Mapping to your application** - This is the dot notation mapping to your FW/1 application. For example, if you placed your application into a folder called *myFW1App* in Mura's *requirements* folder, then the value of this setting would be *myFW1App*.
4. **FW/1 Event Name** - This is the name of the URL or form variable used to specify the desired action in your FW/1 application. The default value in all FW/1 applications is *action*, but this can be overridden by specifying a value for *variables.framework.action*.

Choose which sites you would like the plugin to be available to, and click *Update*. The FW/1 Connector plugin is now installed and ready to be used.

Configuring your FW/1 Application to Work with the Plugin

The following keys must be set in the *variables.framework* structure within the *Application.cfc* file of your application:

1. **applicationKey** - This must be a unique value for each FW/1 application that you wish to embed, and must correspond to the *FW/1 Application Key* specified in the Plugin Settings as discussed above.
2. **base** - This must be the path to the directory containing the layouts and views folders, either as a mapped path or a webroot-relative path (i.e., it must start with */* and expand to a full file system path).

Using the *doEvent* Method

You will use the plugin's *doEvent()* method to fire off events within your FW/1 application. It accepts the following two arguments:

1. **event** - This is Mura's event object.
 2. **action (optional)** - This is the action to be passed to your FW/1 application via a URL variable. If not provided the default action of your FW/1 application will be called.
- Note:** This argument is actually mandatory for now, as there is currently an issue with Mura and FW/1's SES URL scheme. If that issue gets resolved (and I've submitted a patch for it) then this argument will be optional again.

Calling an FW/1 Event from within Mura

As described above, you call an FW/1 event by calling the *doEvent()* method of the FW/1 plugin. The plugin itself will reside in Mura's *event* object, in a key with the same name as specified in the *FW/1 Application Key* setting.

For example, if you used a value of *myFW1App* for the *FW/1 Application Key* setting, then the plugin would be accessible via *event.myFW1App*. This gives you access to the plugin, which allows you to call its *doEvent()* method as required.

The following are two examples of ways in which you might call an FW/1 event. Note that these examples assume that the value of the *FW/1 Application Key* setting is *myFW1App*:

1. By using the Mura tag:

```
[mura]event.myFW1App.doEvent(event, 'main.showLogin') [/mura]
```

2. From within a cfm or component that has access to the Mura event:

```
event.myFW1App.doEvent(event, 'main.showLogin')
```

That's pretty much all there is to it. Now go forth and create plugins!