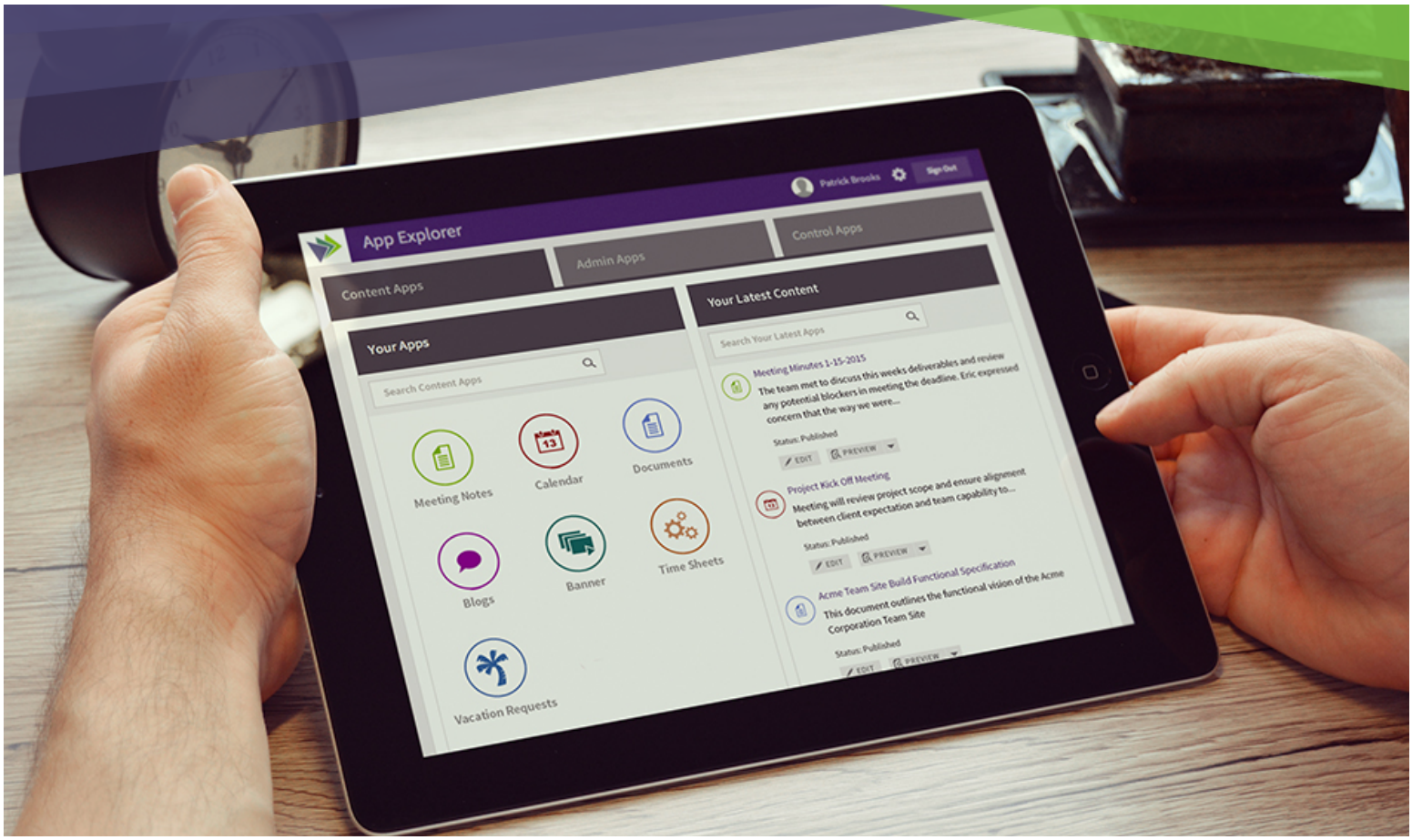


akumina

Akumina Digital Workplace

Page Life Cycle and Custom Steps

Version 1.0 (February 2017)

Table of Contents

Overview

The **Page Life Cycle** is a series of steps that are executed when a user navigates to a page. Steps call a function that can do anything from getting data, populating markup, provide "Loading…" feedback to the user, and updating the User and Configuration Contexts. Step functions have the following structure:

```
function MyCustomStep() {
    //perform step functionality

    //the framework will not go to the next step until you fire this event, so you can get data from an endpoint if needed
    Akumina.Digispace.AppPart.Eventing.Publish('/loader/onexecuted/');

}
```

Context Objects

Steps within the page life cycle can retrieve and update information within the Akumina.Digispace.ConfigurationContext and Akumina.Digispace.UserContext objects.

Akumina.Digispace.ConfigurationContext

The Akumina.Digispace.ConfigurationContext is an object that stores information about the current Digital Workplace Site. The following properties are available to retrieve from OOB.

- **Settings**
- **TemplateFolderName**
- **TemplateCoreFolderName**
- **GraphClientId**
- **GraphSubscriptionId**
- **SiteLogoObj**
- **InterchangeURL**
- **InterchangeLoginURL**
- **CachingStrategyInterval**
- **DepartmentSiteMap**
- **GoogleMapKey**
- **SkypeApiKey**
- **SkypeApiKeyCC**
- **TenantUrl**
- **EnableAzureAD**
- **EnableDebugMode**
- **InterchangeQueryKey**
- **Theme**
- **SearchPageExclusionList**
- **IsMultiLingualEnabled**

Akumina.Digispace.UserContext

The Akumina.Digispace.UserContext is an object that stores information about the current User. The following properties are available to retrieve from OOB:

- **userSharePointGroups**
- **userGroups**
- **userProperties**
- **Department**
- **AdditionalDepartments**
- **DisplayName**
- **City**
- **JobTitle**
- **State**
- **PostalCode**
- **GraphUserId**
- **LoginName**
- **Photo**
- **IsExternal**
- **CanEditPage**
- **IsLiveMode**

Core Steps

The Digital Workplace Framework has the following steps built into its core. The core step names are listed in order of execution.

Auto Clear Local Cache

Clears the local cache while saving the current site id and site url.

Events fired:

- **/loader/onexecuted/**

Fetching Digispace Configuration

Loads the Digispace Configuration values from the DigispaceConfigurationIDS_AK list into the current Akumina.Digispace.ConfigurationContext object.

Events fired:

- **/loader/onexecuted/**

Detect Multiple SiteVisible Languages

Determines whether or not the site has Multilingual enabled and sets a flag within the Akumina.Digispace.ConfigurationContext object accordingly

Events fired:

- **/loader/onexecuted/**

Load User Language Settings

Determines the language to be used on the site from the UserContext. Loads the language specific script.

Events fired:

- **/loader/onexecuted/**

Getting Additional Markup template

Loads **AdditionalMasterMarkup.html** and appends it to the page.

Events fired:

- **/loader/onexecuted/**

Display Loader

Checks if there are any cached steps. If not, displays the loader icon.

Events fired:

- **/loader/showloading/**
- **/loader/onexecuted/**

Event Subscription

The **/genericlistcontrol/loaded/** event is subscribed and will render widgets on the dashboard when fired.

Events fired:

- **/loader/eventsubscription/**
- **/loader/onexecuted/**

Set Site Theme

Appends the selected theme css to the page.

Events fired:

- **/loader/onexecuted/**

Validating Interchange Query Key

Validates the Interchange Query Key.

Events fired:

- **/loader/onexecuted/**

Get Loading Template

Retrieves **Loading.html** and saves it within the Akumina.Digispace.ConfigurationContext object

Events fired:

- **/loader/onexecuted/**

Index Page Data

Reindexes the PageData_AK list

Events fired:

- **/loader/onexecuted/**

Getting Interchange LoginURL

Retrieves the Interchange Login URL and sets it to the cache.

Events fired

- **/loader/onexecuted/**

Getting Graph Token

If Azure AD is enabled, the Graph Token is retrieved.

Events fired:

- **/loader/onexecuted/**

Fetching User Properties

If Azure AD is enabled, fetch user properties. Else, just fetch the page level permissions

Events fired:

- **/loader/onexecuted/**

Load Widgets

Loads properties for widget instances from WidgetProperties_AK

Events fired:

- **/loader/onexecuted/**

Initialize Widgets

Initializes widget instances on the page.

Events fired:

- **/loader/onexecuted/**

Initialize Generic Controls

Calls init functions for OOB widgets

Events fired:

- **/loader/onexecuted/**

Load Dashboard Widgets

Loads Dashboard Widgets from **DashboardWidgets_AK** and initializes them

Events fired:

- **/loader/onexecuted/**

Display Sharepoint Bar

Displays or Hides SharePoint Bar depending on a flag

Events fired:

- **/loader/handlesharepointbar/**
- **/loader/onexecuted/**

Debug Info

Binds the Debug window to **ctrl+up** and the Widget Manager to **ctrl+down**

Events fired:

- **/loader/onexecuted/**

Init Tray

Initializes Akumina Editing buttons in the bottom left corner

Events fired:

- **/loader/onexecuted/**

Digital Workplace Shipped Site Steps

Getting Footer Markup template

Retrieves **FooterMarkup.html** and appends it to the footer of the Digital Workplace Master Page.

Added after **Load User Language Settings**

Events fired:

- **/loader/onexecuted/**

Display Site Logo

Retrieve the Site logo image and add it to the master page.

Added after **Get Loading Template**

Events fired:

- **/loader/onexecuted/**

Initializing Rail

Retrieves Rail Items from Cache or **Rail_AK** list if the cache is empty.

Added after **Fetching User Properties**

Events fired:

- **/loader/onexecuted/**

Initialize Search Fields

Bind the Search Fields within the Digital Workplace Master Page to the Site Search

Added after **Load Dashboard Widgets**

Events fired:

- **/loader/onexecuted/**

Set Department Site Color

Retrieves the Department color and sets the department css to use it.

Will only run in a subsite. Added after **Getting Interchange LoginURL** if the user is logged in, else it is added after **Index Page Data**

Events fired:

- **/loader/onexecuted/**

Adding Custom Steps

Akumina.Digispace.AddStepsAfter

In our code, we can add a custom step after a core step using the format below

```
Akumina.Digispace.Loader.AddStepsAfter("<CoreStepName>", [{ name: "<NewStepName>",
callback: <NewStepFunction> }]);
```

<CoreStepName>

Replace this with the name of the Core Step you are adding after, ie “Auto Clear Local Cache”. They are listed under the Core Steps section.

<NewStepName>

Replace this with the name of your new step

<NewStepFunction>

Replace this with the name of the function that executes your step code.

Chaining Steps

```
Akumina.Digispace.Loader.AddStepsAfter("Auto Clear Local Cache", [{ name: "My First
Step", callback: MyFirstStep }]);


Akumina.Digispace.Loader.AddStepsAfter("Auto Clear Local Cache", [{ name: "My Second
Step", callback: MySecondStep }]);
```

The situation above where we add two new steps after the same step *will not* work. When this happens the second step added will overwrite the first. The correct way to handle this situation is to “chain” these steps, by adding the second step after the newly added first step, as shown below.

```
Akumina.Digispace.Loader.AddStepsAfter("Auto Clear Local Cache", [{ name: "My First
Step", callback: MyFirstStep }]);


Akumina.Digispace.Loader.AddStepsAfter("My First Step", [{ name: "My Second Step",
callback: MySecondStep }]);
```

Note: If you are doing customizations on a Digital Workplace Site you must take note of the Shipped Site Steps when adding new custom step. These steps are incorporated by **Akumina.Digispace.Loader.AddStepsAfter** and you need to be careful not to overwrite them when adding additional steps.

AdditionalSteps Object

In our digitalworkplace.custom.js we will need to create an AdditionalSteps object that will initialize our custom steps in the Page Life Cycle.

```
var AdditionalSteps = AdditionalSteps || {}

if ((typeof AdditionalSteps.MoreSteps) === 'undefined') {

    AdditionalSteps.MoreSteps = {

        Init: function () {
                  console.log('AdditionalSteps.MoreSteps.Init');
      //Add Custom Steps Here
        }
    }
}
```

var AdditionalSteps = AditionalSteps || {}

This line calls the Init function of all objects grouped under AdditionalSteps

if ((typeof AdditionalSteps.MoreSteps) === 'undefined')

This defines the AdditionalSteps.MoreSteps object. We can create additional objects under AdditionalSteps called whatever we want, ie AdditionalSteps.MyAdvancedSteps, AdditionalSteps.HelloWorld, AdditionalSteps.ProgrammingIsFun, and their Init functions will all be called

Init: function ()

This is Init function for the AdditionalSteps.MoreSteps object. Here is where we want to add our custom steps.

Custom Step Samples

Hello World Custom Step

We will create a custom function and add it as a custom step. When the Page Life Cycle reaches the step the alert will fire.

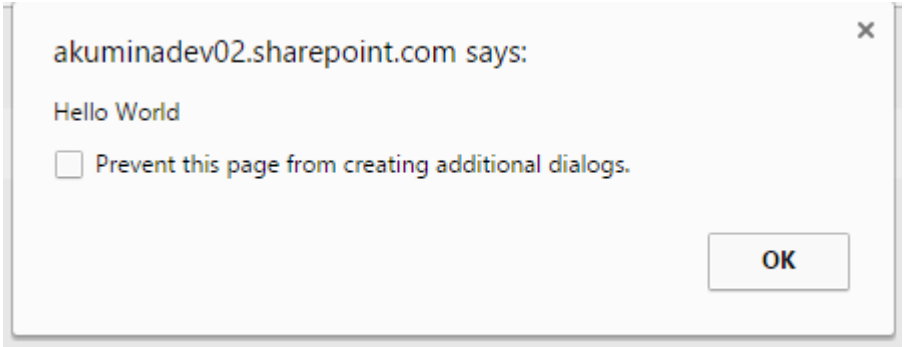
How to Deploy

1. **Download digitalworkplace.custom.js from the "/Style Library/DigitalWorkplace/JS" folder within SharePoint**
2. **Paste the Code within digitalworkplace.custom.js**
3. **Upload the updated digitalworkplace.custom.js to the "/Style Library/DigitalWorkplace/JS" folder within SharePoint**
4. **Navigate to a page on your Digital Workplace site**
5. **Flush your cache by clicking on the Akumina icon in the left rail, clicking the refresh icon, then clicking "Refresh All Cache"**
6. **Refresh the page. The alert should fire.**

Code

```
var AdditionalSteps = AdditionalSteps || {
}
if ((typeof AdditionalSteps.MoreSteps) === 'undefined') {
    AdditionalSteps.MoreSteps = {

        Init: function () {
            console.log('AdditionalSteps.MoreSteps.Init');
            Akumina.Digispace.Loader.AddStepsAfter("Auto Clear Local Cache", [{ name:
"Hello World Step", callback: HelloWorld }]);
        }
    }
}

function HelloWorld() {
    alert("Hello World");
    Akumina.Digispace.AppPart.Eventing.Publish('/loader/onexecuted/');
}
```

Result



Accessing UserContext

We will create a custom step that retrieves the login name of the user as well as a custom step that stores data in a custom property within the UserContext. We will chain these steps.

How to Deploy

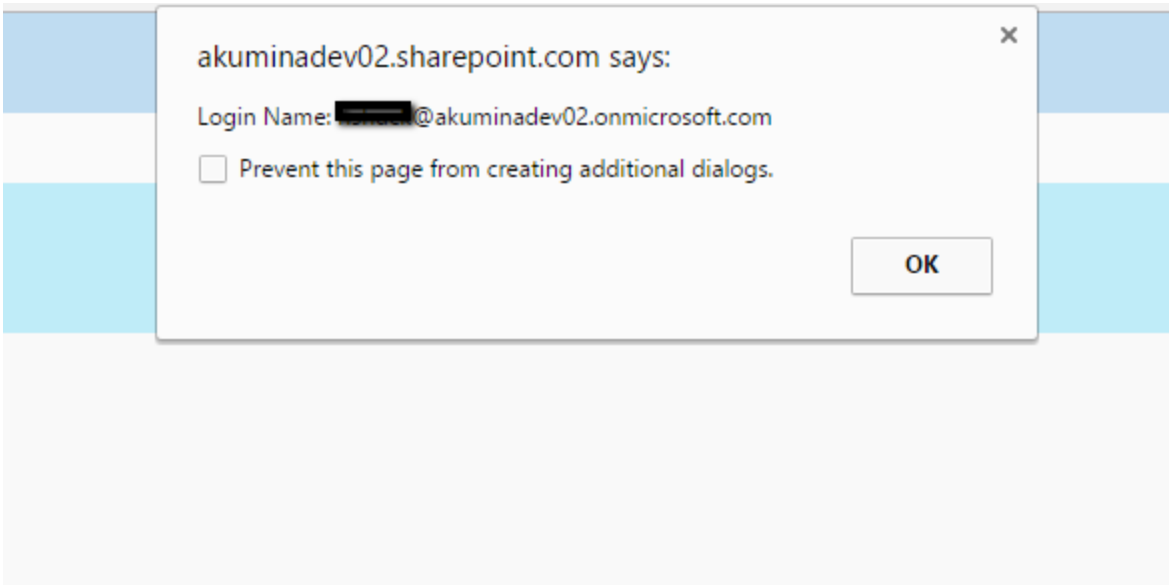
1. **Download digitalworkplace.custom.js from the "/Style Library/DigitalWorkplace/JS" folder within SharePoint**
2. **Paste the Code within digitalworkplace.custom.js**
3. **Upload the updated digitalworkplace.custom.js to the "/Style Library/DigitalWorkplace/JS" folder within SharePoint**
4. **Navigate to a page on your Digital Workplace site**
5. **Flush your cache by clicking on the Akumina icon in the left rail, clicking the refresh icon, then clicking "Refresh All Cache"**
6. **Refresh the page. The alert should fire. Inspect the page and enter Akumina.Digispace.UserContext.MyCustomProperty into the console to view your property**

Code

```
var AdditionalSteps = AdditionalSteps || {
}

if ((typeof AdditionalSteps.MoreSteps) === 'undefined') {

    AdditionalSteps.MoreSteps = {

        Init: function () {

            console.log('AdditionalSteps.MoreSteps.Init');

            Akumina.Digispace.Loader.AddStepsAfter("Initializing Rail", [{ name: "Access
User Context", callback: AccessUserContext }]);
                    Akumina.Digispace.Loader.AddStepsAfter("Access User Context", [{ name:
"Add My Data To User Context", callback: AddMyDataToUserContext }]);
```

```
        }
    }
}

function AccessUserContext() {
      //Grab LoginName
      var loginName = Akumina.Digispace.UserContext.LoginName;

      //Display LoginName
      alert ("Login Name: " + loginName);

      //Move on to next step
      Akumina.Digispace.AppPart.Eventing.Publish('/loader/onexecuted/');

}
function AddMyDataToUserContext() {
    //stuff custom data into Digispace usercontext
    Akumina.Digispace.UserContext.MyCustomProperty = "Custom Value";

      //Move on to next step
    Akumina.Digispace.AppPart.Eventing.Publish('/loader/onexecuted/');

}
```

Result



After the site is deployed. Inspect the page and enter **Akumina.Digispace.UserContext.MyCustomProperty** into the console

```
> Akumina.Digispace.UserContext.MyCustomProperty
< "Custom Value"
```

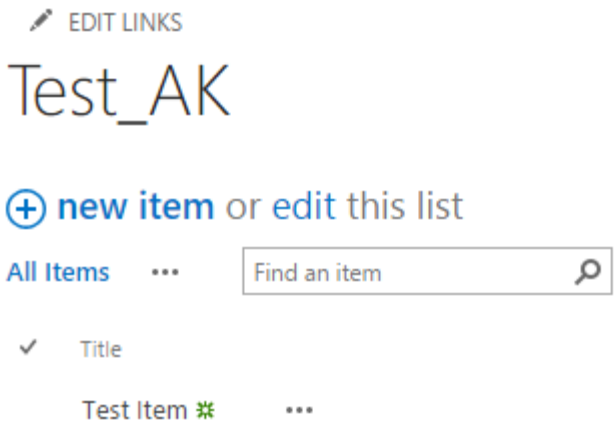
Accessing a List

We will create a custom step that retrieves the first value in the **Test_AK** list.

How to Deploy

1. **Create a custom SharePoint list called "Test_AK". No additional columns are required. Add one item.**
2. **Download digitalworkplace.custom.js from the "/Style Library/DigitalWorkplace/JS" folder within SharePoint**
3. **Paste the Code within digitalworkplace.custom.js**
4. **Upload the updated digitalworkplace.custom.js to the "/Style Library/DigitalWorkplace/JS" folder within SharePoint**
5. **Navigate to a page on your Digital Workplace site**
6. **Flush your cache by clicking on the Akumina icon in the left rail, clicking the refresh icon, then clicking "Refresh All Cache"**
7. **Refresh the page. The alert should fire.**

List



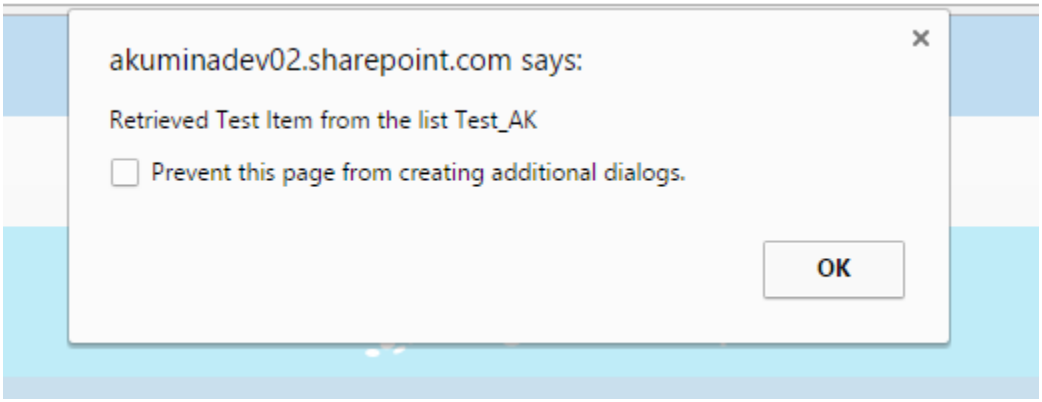
Code

```
var AdditionalSteps = AdditionalSteps || {
}
if ((typeof AdditionalSteps.MoreSteps) === 'undefined') {

    AdditionalSteps.MoreSteps = {

        Init: function () {
```

```
            console.log('AdditionalSteps.MoreSteps.Init');
                  Akumina.Digispace.Loader.AddStepsAfter("Debug Info", [{name: "Access a
List", callback: RetrieveFromList }]);

        }

    }

}

function RetrieveFromList()
{
      var listName = "Test_AK";

      var request = {};
      request.listName = listName;
      request.selectFields = 'Title';
      request.rowLimit = 100;

      var spcaller = new Akumina.Digispace.Data.SharePoint();
      spcaller.GetList(request).then(function (data) {
            var request = data.request;
            var response = data.response;
            var listItems = response.listItems;

            var items = [];
            var listEnumerator = listItems.getEnumerator();
            var count = 0;
            while (listEnumerator.moveNext()){
                  var listItem = listEnumerator.get_current();
                  var title = listItem.get_item('Title');
                  items.push({ title: title});
            }

            alert("Retrieved " + items[0].title + " from the list " + request.listName);
            Akumina.Digispace.AppPart.Eventing.Publish('/loader/onexecuted/');

      }, function (error) {
            var args = error.args;
```

```
            alert ("Request failed. " + args.get_message() + "\n" +
args.get_stackTrace());
            Akumina.Digispace.AppPart.Eventing.Publish('/loader/onexecuted/');
      });
}
```

Result



Appending Additional Markup

We will create a custom step that adds markup to the footer of a Digital Workplace Page.

How to Deploy

1. **In the Management Apps tab of Interchange, click on the View Manager. Click "Add New". In the left pane navigate to "/DigitalWorkplace/Content/Templates/" for the folder path. Click "Choose File", navigate to your custom template (CustomTemplate.html). Click "Save".**
2. **Download digitalworkplace.custom.js from the "/Style Library/DigitalWorkplace/JS" folder within SharePoint**
3. **Paste the Code within digitalworkplace.custom.js**
4. **Upload the updated digitalworkplace.custom.js to the "/Style Library/DigitalWorkplace/JS" folder within SharePoint**
5. **Navigate to a page on your Digital Workplace site**
6. **Flush your cache by clicking on the Akumina icon in the left rail, clicking the refresh icon, then clicking "Refresh All Cache"**
7. **Refresh the page. The custom markup will be added to the footer.**

Template

Create a file called CustomTemplate.html and paste the following code within

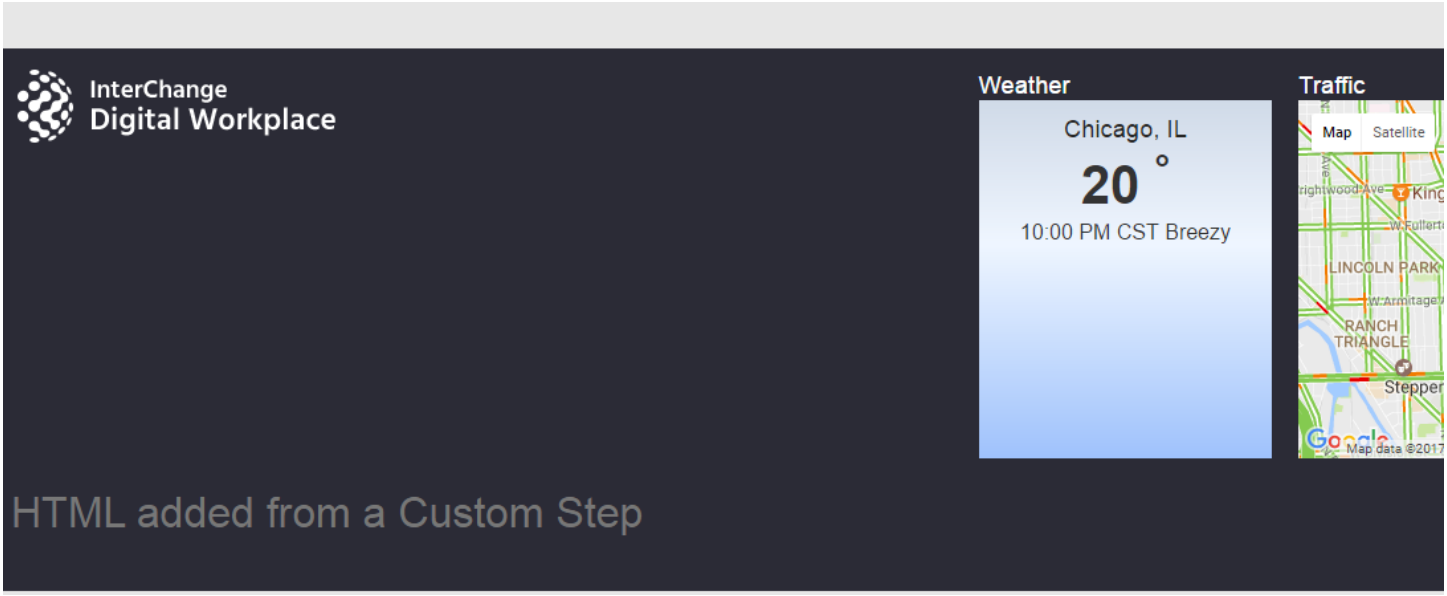
```
<div>
      <h1>HTML added from a Custom Step</h1>
</div>
```

Custom Step

```
var AdditionalSteps = AdditionalSteps || {
}
if ((typeof AdditionalSteps.MoreSteps) === 'undefined') {
```

```
    AdditionalSteps.MoreSteps = {

        Init: function () {

                console.log('AdditionalSteps.MoreSteps.Init');

      Akumina.Digispace.Loader.AddStepsAfter("Getting Additional Markup template",
[{name: "Add Custom Template", callback: AddCustomTemplate }]);

        }

    }

}

function AddCustomTemplate() {
      var popupTemplateUrl = __getTemplatePrefix() + "/Style%20Library/" +
Akumina.Digispace.ConfigurationContext.TemplateCoreFolderName +
"/Content/Templates/CustomTemplate.html";

            new
Akumina.Digispace.AppPart.Data().Templates.ParseTemplate(popupTemplateUrl,
{}).done(function (html) {
                $("#digiFooter").append(html);
                Akumina.Digispace.AppPart.Eventing.Publish('/loader/onexecuted/');
            });
}
```

Result


InterChange
Digital Workplace
Weather
Chicago, IL
20 °
10:00 PM CST Breezy
Traffic
Map
Satellite
LINCOLN PARK
RANCH TRIANGLE
Stepper
Google
Map data ©2017
HTML added from a Custom Step