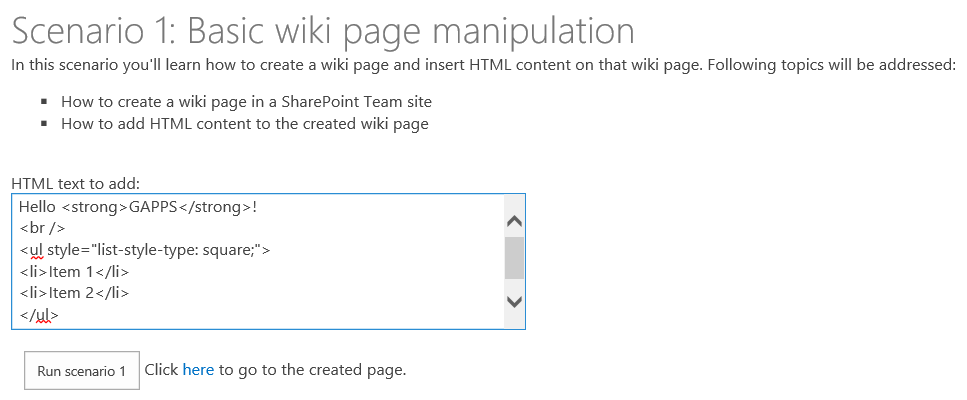
# Office AMS: Page manipulation

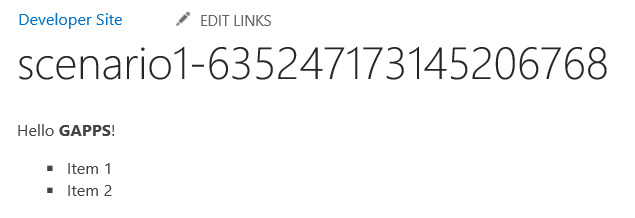
|  |  |
| --- | --- |
| Summary: | Applies to: |
| This sample shows how to add wiki pages to a SharePoint site, how to add/remove web parts and HTML content to the created wiki pages | * Office 365 Multi Tenant (MT) * Office 365 Dedicated (D) * SharePoint 2013 on-premises |
| Solution: | Provisioning.Pages, version 2.0 |
| Author: | Bert Jansen (Microsoft) |
| //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  // THIS CODE IS PROVIDED \*AS IS\* WITHOUT WARRANTY OF  // ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING ANY  // IMPLIED WARRANTIES OF FITNESS FOR A PARTICULAR  // PURPOSE, MERCHANTABILITY, OR NON-INFRINGEMENT.  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* | |

# Scenario 1: Basic Wiki Page Manipulation

This scenario shows two basic steps: first step is creating a wiki page in a wiki library. The sample uses the SitePages library which is default available in non-publishing sites, but the same sample can be easily adapted to create pages in a custom wiki page library. In the last step the empty wiki page is filled with HTML content. In the sample you can specify your HTML content and then click the “Run scenario 1” button to create the wiki page and add the HTML:



Once the page has been created the “here” link points to the created page. Clicking on the link brings you to the created page:



## Create wiki page

To create a wiki page the following code is used:

string scenario1Page = String.Format("scenario1-{0}.aspx", DateTime.Now.Ticks);

string scenario1PageUrl = csomService.AddWikiPage("Site Pages", scenario1Page);

## Adding HTML to a wiki page

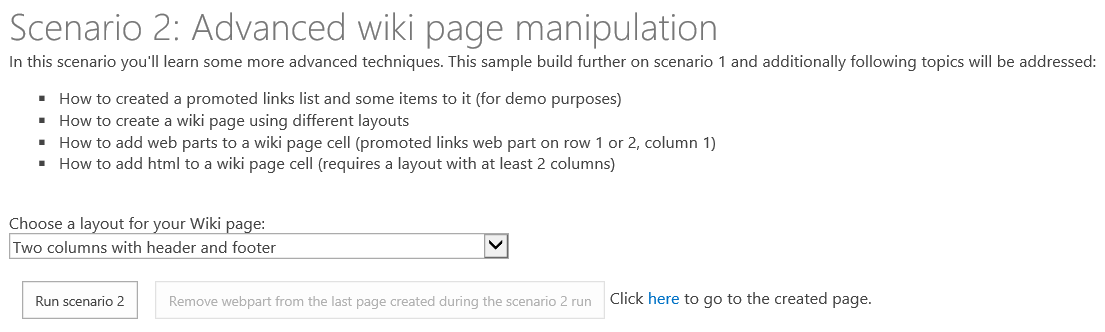
This is simply done by using the extension method AddHtmlToWikiPage from Office AMS Core:

cc.Web.AddHtmlToWikiPage("SitePages", txtHtml.Text, scenario1Page);

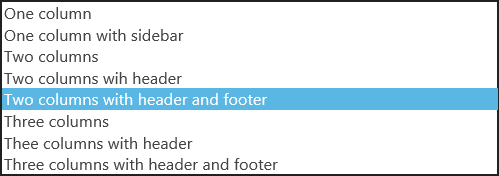
Adding HTML to a wiki page is simple: just put your HTML in the WikiField cell:

listItem["WikiField"] = html;

# Scenario 2: Advanced Wiki Page manipulation



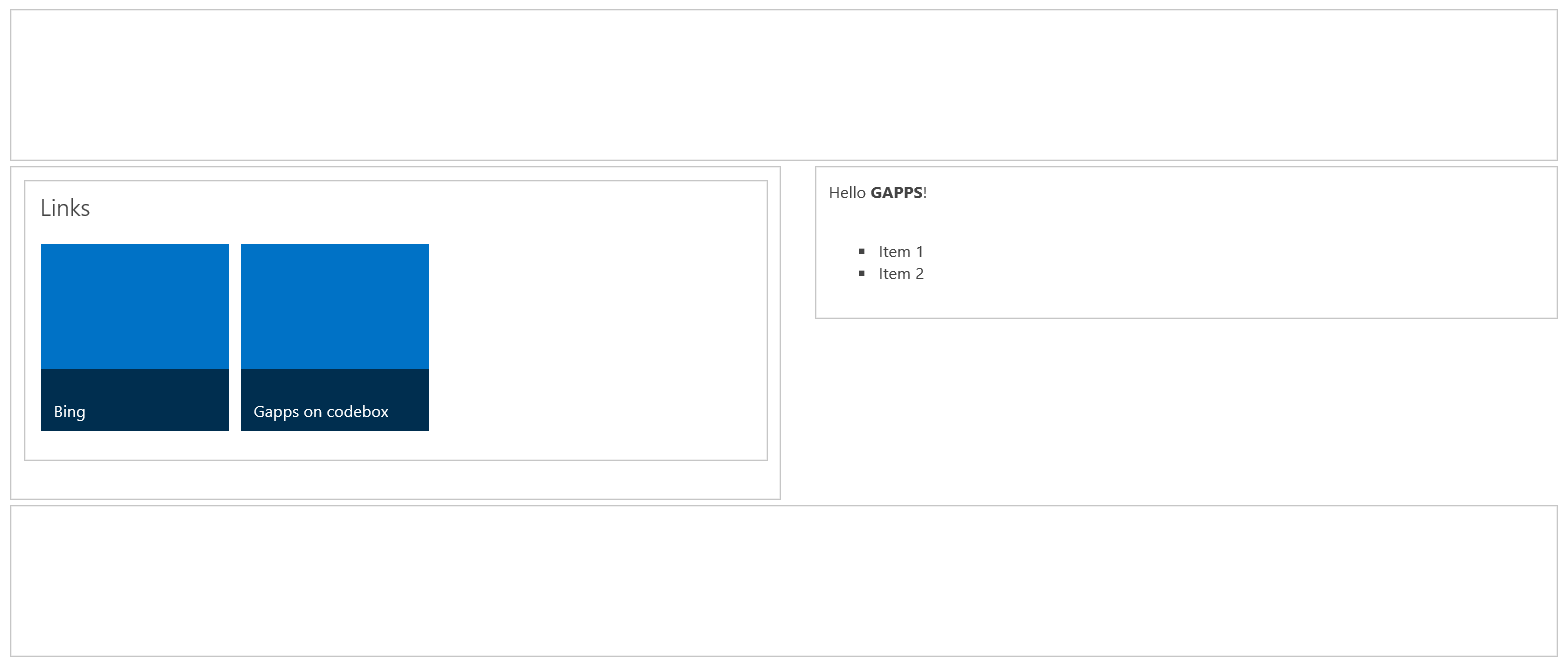
This scenario further extends the scenario 1 with the option to create wiki pages using different layouts:



In the sample the existing OOB layouts are reused, but it’s perfectly possible to use custom layouts as well (e.g. a “four columns” layout). Once the layout is created we have actually created a table with rows and columns and optionally headers and footers. In a next step the sample shows how you can place either a web part or HTML content into a cell of the created table. When you click on “Run scenario 2” a wiki page will be created with the layout you’ve selected and:

* An XsltListViewWebPart is placed in the first content row on column 1. If there’s no header this means row1, column 1. If there’s a layout with a header then this will be row 2, column 1. The XsltListView web part is created for a promoted links list which will be automatically created if it does not yet exist
* If there are two or more colums than the HTML content listed for scenario 1 will be inserted in the column next to the web part

Once the page is created you can use the here link to show the created page. Below screenshot shows the page in edit mode so that you can see the used layout:



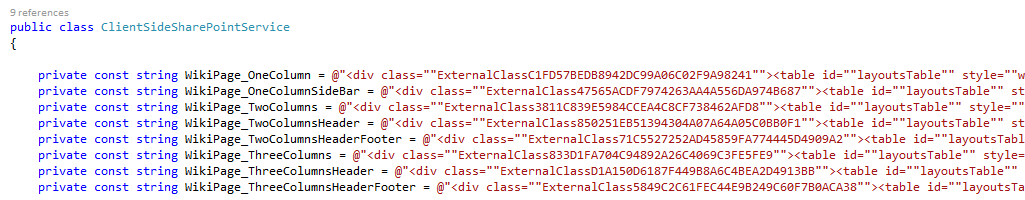
Once a page has been created the button “Remove webpart from the last page created during scenario 2 run” is enabled. Clicking on this button removes the XsltListViewWebPart web part from the page.

## Apply layouts to a wiki page

Below sample shows how to insert a two columns with header and footer layout into a created wiki page:

cc.Web.AddLayoutToWikiPage("SitePages", WikiPageLayout.TwoColumnsHeaderFooter, scenario2Page);

Technically this is nothing more than inserting the correct HTML table structure as HTML content. The HTML table structure for the OOB layouts is included in the Office AMS Core library:



## Inserting a web part

To insert a web part you need to create a webpartentity class instance, fill it with the web part data and insert it:

Guid linksID = csomService.GetListID("Links");

WebPartEntity wp2 = new WebPartEntity();

wp2.WebPartXml = WpPromotedLinks(linksID, string.Format("{0}/Lists/{1}", Request.QueryString["SPHostUrl"], "Links"), string.Format("{0}/{1}", Request.QueryString["SPHostUrl"], scenario2PageUrl), "$Resources:core,linksList");

wp2.WebPartIndex = 1;

wp2.WebPartTitle = "Links";

cc.Web.AddWebPartToWikiPage("SitePages", wp2, scenario2Page, 1, 1, false);

1, 1 in the above sample tell the method to insert on row 1, column 1.

In above sample the webpart XML for the promoted links list is adapted to work for the created list:



Note:

The demo does not show how to add a web part to a layouts page, but this is possible using the AddWebPartToWebPartPage method:

public void AddWebPartToWebPartPage(WebPartEntity webPart, string page)

## Inserting HTML to a wiki page cell

cc.Web.AddHtmlToWikiPage("SitePages", txtHtml.Text, scenario2Page, 1, 2);

1, 2 in the above sample tell the method to insert on row 1, column 2.

## Deleting a web part

To delete a web part use the following code:

cc.Web.DeleteWebPart("SitePages", "Links", "LastPageName.aspx");

This method deletes the Links webpart from the page LastPageName.aspx which is stored in the wiki page library “SitePages”.

Note:

This method can also be used to remove a web part from a layouts page. In this case you should specify an empty library name:

cc.Web.DeleteWebPart("", "About this blog", "Default.aspx");

This sample removes the “About this blog” web part from the blog site home page.

# Scenario cleanup

If you play around with this sample you’ll have created a collection of pages in the sitepages library. Clicking on the “Cleanup created pages” button removes all these pages. Note that the created Promoted Links list will not be automatically deleted.

