## Developing App Parts in a SharePoint-hosted App

**Lab Time**: 40 minutes

**Lab Folder**: C:\Student\SharePointHostedApps\Lab

**Lab Overview**: In this lab you will create a new SharePoint-hosted App project and get some experience developing and testing custom app parts.

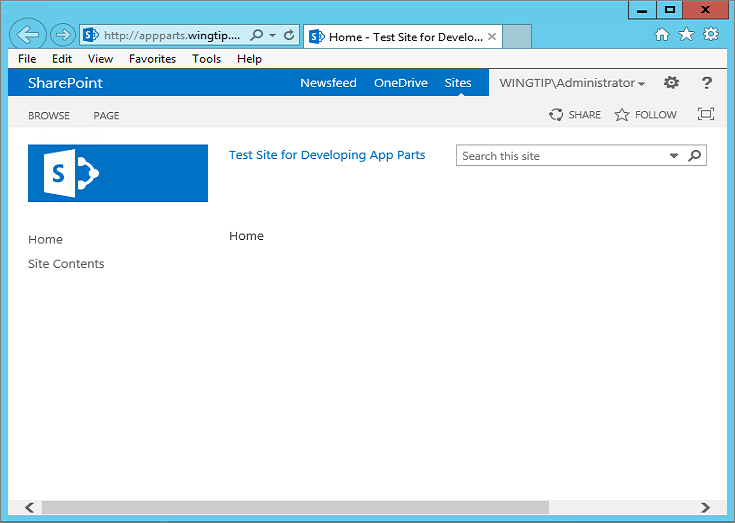
### Exercise 1: Creating the Hello World App Part

In this exercise you will create a new SharePoint-Hosted App with a simple app part.

1. Create a new **Blank site** for testing in this lab:
   1. Ensure you are logged into the **WingtipServer** server as **WINGTIP\Administrator**.
   2. Run a PowerShell script, found in the root lab folder for this module:
      1. Right-click **SetupLab.ps1** and select **Run with PowerShell**. This file can be found in the files associated with this lab:  
         (Note: if prompted to change the execution policy for PowerShell press **Y** and **Enter**.

C:\Student\Modules\SharePointHostedApps\Lab

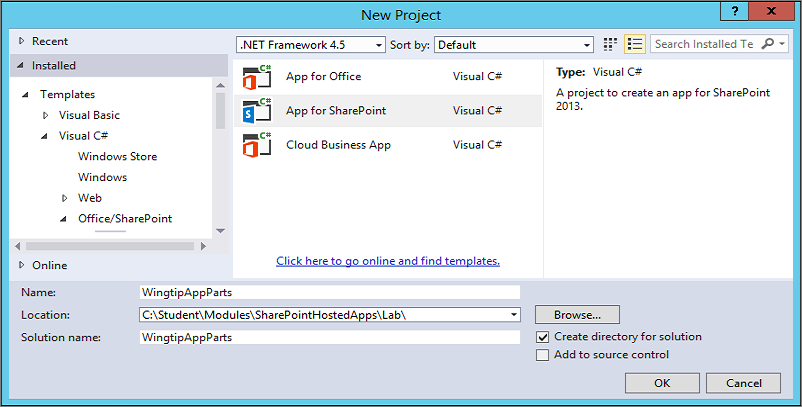
* 1. When the script completes, it will launch a new browser and navigate to site at **http://apppart.wingtip.com**.



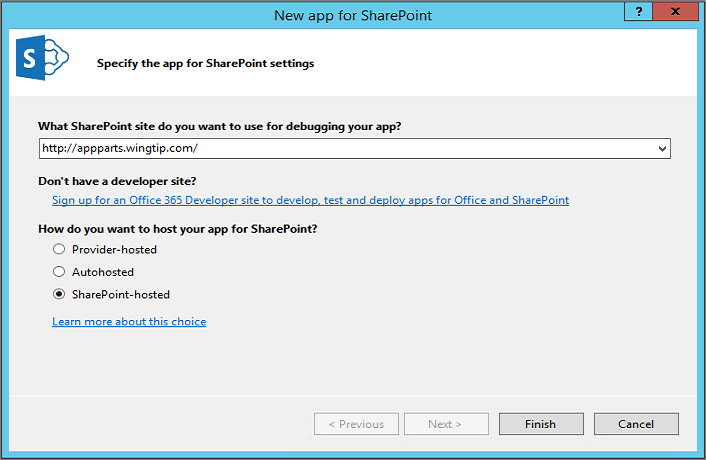
Note that this site has been created using the Blank site template. A local Blank site can make testing app parts it a little easier because it has a simple web part page as its home page whereas the home page for Team sites and Developer sites as wiki pages which can add unneeded complexity.

* 1. Close the PowerShell console window.

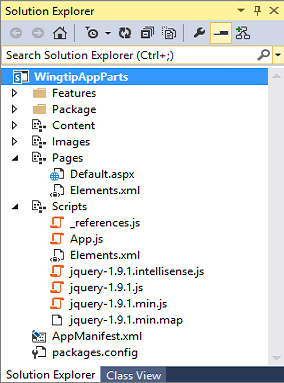
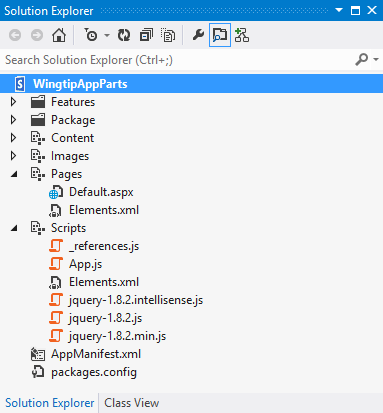
1. Create a new project in Visual Studio 2012:
   1. Launch **Visual Studio 2012** as administrator:
      1. Windows Keyboard Key 🡪 Right click on the Visual Studio 2012 tile and select Run as administrator.
   2. In Visual Studio select **File 🡪 New 🡪 Project**.
   3. In the **New Project** dialog:
      1. Find the **App for SharePoint** template under the **Templates 🡪 Visual C# 🡪 Office / SharePoint 🡪 Apps** section.
      2. **Name**: WingtipAppParts
      3. **Location:** C:\Student\Modules\SharePointHostedApps\Lab
      4. Click **OK.**



* 1. **In the New app for SharePoint dialog, enter the information as shown in the following screenshot which use the test site that you created earlier this lab at http://appparts.wingtip.com.**



* 1. **Once the new project has been created, examine its structure and the source files inside.**



1. **Open the AppManifest.xml file by double clicking on it. Change the Title to Wingtip App Parts. Save and close the AppManifest.xml file.**
2. **Open the App.js file in the Scripts folder. Delete all the contents from App.js leaving it as an empty file for now. Save your changes to App.js and close the file.**
3. **Open Default.aspx in Code View. Replace the content inside the two placeholders named PlaceHolderPageTitleInTitleArea and PlaceHolderMain with the following HTML code.**

<asp:Content ContentPlaceHolderID="PlaceHolderPageTitleInTitleArea" runat="server">

****Wingtip App Parts****

</asp:Content>

<asp:Content ContentPlaceHolderID="PlaceHolderMain" runat="server">

****<h2>Wingtip App Parts</h2>****

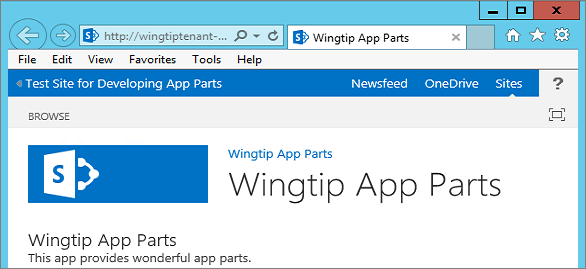
****<div>This app provides wonderful app parts.</div>****

</asp:Content>

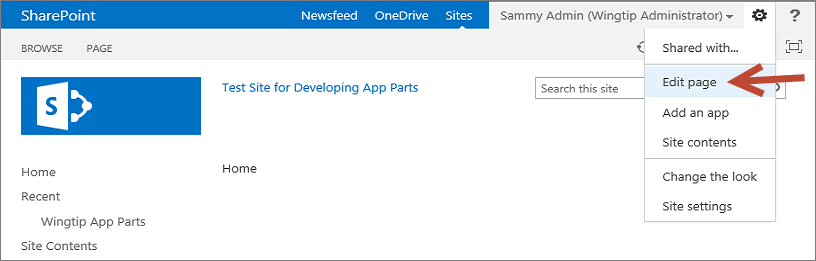
1. **Save your changes to Default.aspx and then close this file.**

**Remember that the purpose of this lab exercise is to create app parts. However, the app still requires a start page even if the start page doesn’t really provide any real functionality. However, the start page is helpful for testing because it provides a link back to the host web where you will be testing and debugging your app parts. When you launch a debugging session, you should become familiar with the process of redirecting from the app start page back to the host web so you can create an instance of your app parts for testing and debugging purposes.**

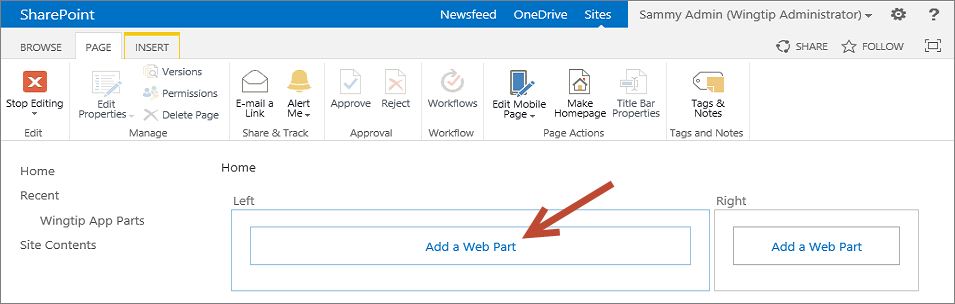
1. **Test your work by pressing the {F5} key to launch a debugging session. When the app starts, you should see the start page appear as the one shown in the following screenshot. Leave this start page open for the next step.**



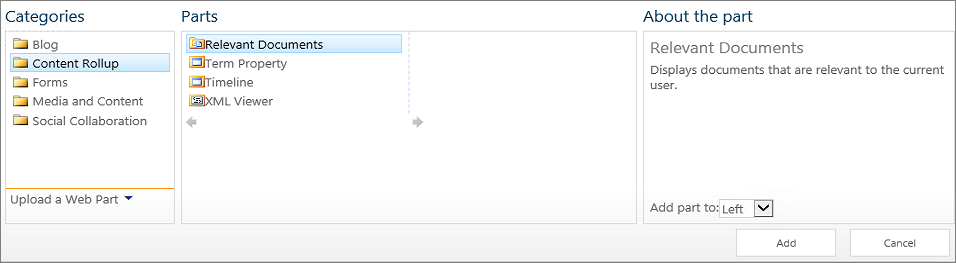
1. **Navigate to the host web and see what web parts are available out of the box.**
   1. **Click the link in the top left corner of the start page to navigate back to the host web. This should redirect you to the home page of the Blank site at http://apppart.wingtip.com.**
   2. **Drop down the Site Actions menu and select the Edit page command to move the page into Edit Mode.**



* 1. **Once the page is in Edit Mode, you should see two web part zones. Click the Add a Web Part link in the left web part zone. This action will display the web part catalog.**

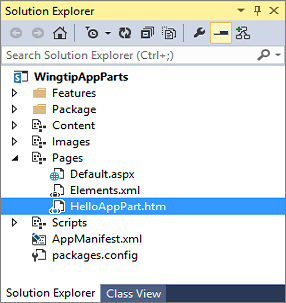


* 1. **While you do not need to add a web part to the page in this step, your objective is simply to see what web parts are available. In just a bit, your will see your custom app parts available in this web part catalog.**

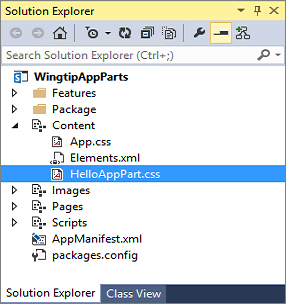


* 1. **Once you have looked through the available set of out-of-the-box web parts, close the browser to end the debugging session.**

1. **Return to Visual Studio.**
2. **Create an HTML page for an app part.**
   1. **Add a new HTML page to the Pages folder named HelloAppPart.htm.**
      1. **In the WingtipAppParts project right click on the Pages folder and select Add 🡪 New Item…**
      2. In the **Add New Item** dialog box, Select **Visual C# 🡪 Web** from the categories on the left side then select **HTML Page** from the templates in the middle and give this page the name: **HelloAppPart.htm**



* 1. **Add a new CSS file to the Content folder named HelloAppPart.css.**
     1. **In the WingtipAppParts project right click on the Content folder and select Add 🡪 New Item…**
     2. In the **Add New Item** dialog box, Select **Visual C# 🡪 Web** from the categories on the left side then select **Style Sheet** from the templates in the middle and give this page the name: **HelloAppPart.css**



* 1. **Modify the contents of HelloAppPart.css to look like the following CSS listing.**

body {

****background-color: yellow;****

}

****h4 {****

****color: blue;****

****border-bottom: 1px solid blue;****

****}****

* 1. **Save and close HelloAppPart.css.**
  2. **Open HelloAppPart.htm and modify the HTML contents to look like the following HTML listing. Be sure to include a link to the CSS file named HelloAppPart.css.**

<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8" />

<title></title>

****<link href="../Content/HelloAppPart.css" rel="stylesheet" />****

</head>

<body>

****<h4>Hello App Part Content</h4>****

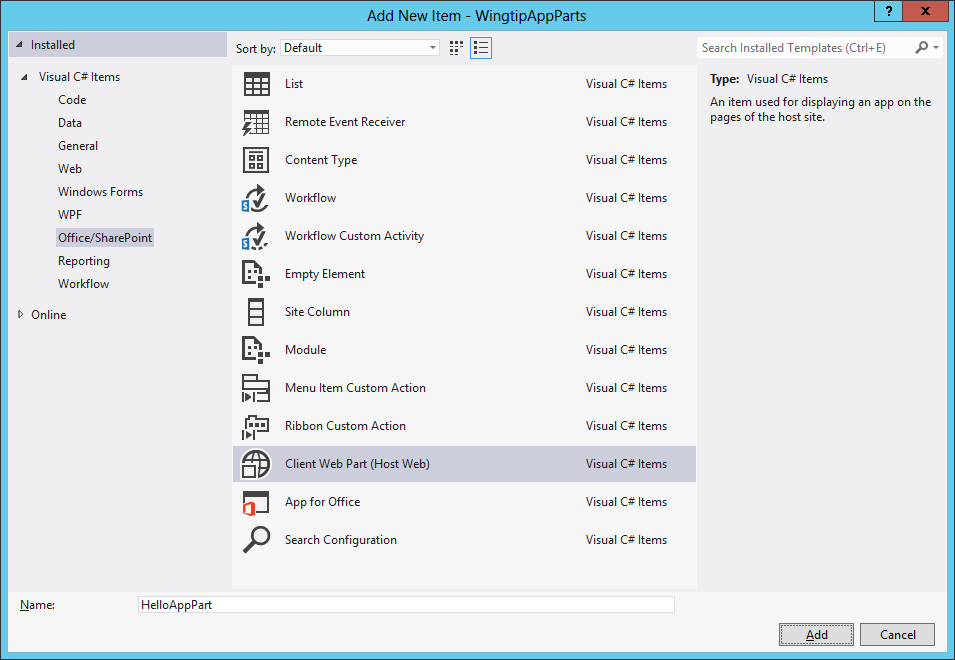
****<div>This content lives in the app web</div>****

</body>

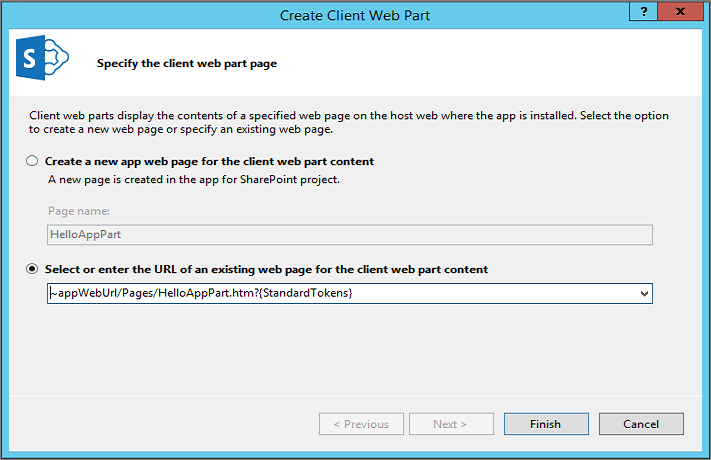
</html>

* 1. **Save and close HelloAppPart.htm.**

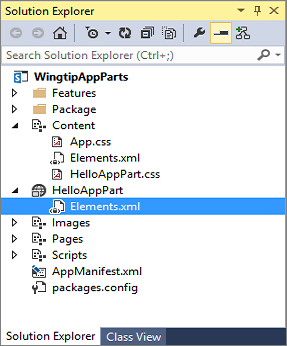
1. **Create a new app part which will use the page HelloAppPart.htm to display its contents.**
   1. **In the Solution Explorer, right-click on the WingtipAppParts project and select the Add New Item command.**
   2. **In the Add New Item dialog, select the Client Web Part (Host Web) project item template and give it the name HelloAppPart.**



* 1. **Click the Add button at the bottom right of the Add New Item dialog to add the new Client Web Part project item. When you click the Add button, you should then see the Create Client Web Part dialog.**
  2. **In the Create Client Web Part dialog, select the option Select or enter a URL for an existing web page. Then use the drop down list to select the HelloAppPart.htm page in the Pages folder.**



* 1. **Click the Finish button in the Create Client Web Part dialog to complete the process of adding the new Client Web Part project item.**
  2. **Once the Client Web Part project item has been created, you can see that Visual Studio has created a folder for it in the project. This folder contains a file named elements.xml.**



* 1. **Modify the elements.xml file for the new HelloAppPart Client Web Part to match the XML in the following code listing.**

<?xml version="1.0" encoding="utf-8"?>

<Elements xmlns="http://schemas.microsoft.com/sharepoint/">

<ClientWebPart

Name="HelloAppPart"

Title=****"The Hello App Part"****

Description=****"A simple little app part"****

DefaultWidth="****600"**** DefaultHeight="200">

<Content

Type="html"

Src="~appWebUrl/Pages/HelloAppPart.htm?{StandardTokens}" />

<Properties>

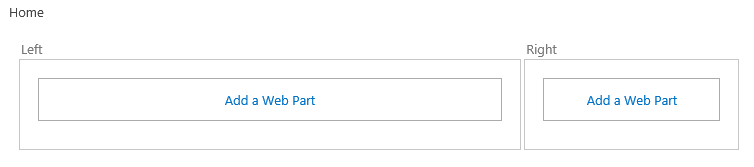
</Properties>

</ClientWebPart>

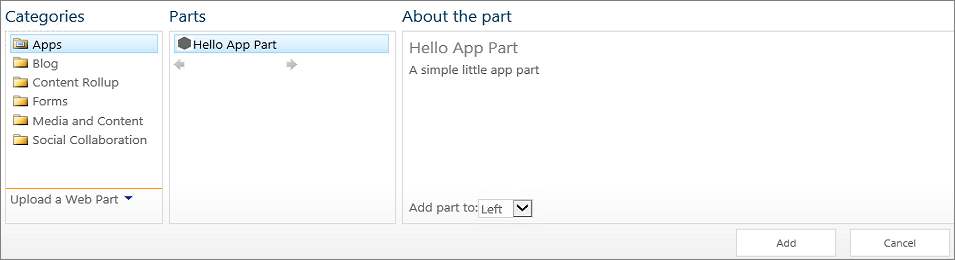
</Elements>

* 1. **Save and close the elements.xml file.**

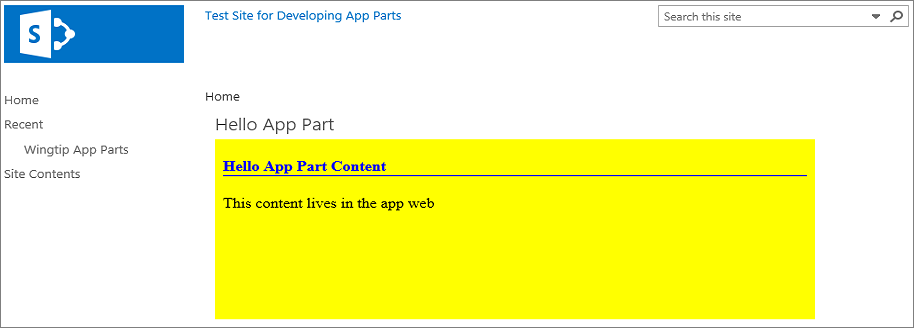
1. **Test your work by adding the HelloAppPart app part to a web part page in the host web.**
   1. **Press {F5} to begin a debugging session.**
   2. **When you see the app's start page, click the link to redirect to the home page of the host web.  
      (Reminder: this link is in the top left corner of the page)**
   3. **Use the Edit page menu from the Site Actions menu to move the page into Edit Mode.**
   4. **Once you are in Edit Mode, click the Add a Web Part link in the left web part zone to display the web part catalog.**



* 1. **Locate and select the app part with a title of The Hello App Part in the Apps category folder. Click the Add button on the bottom right-hand side of the web part catalog to add the app part to the home page of the host web.**



* 1. **In the Ribbon Bar Page Tab click Stop Editing. Now click the Browse Tab in the Ribbon to see your completed page with the app part.**
  2. **After you have added the app part, you should be able to see it on the home page of the host web.**



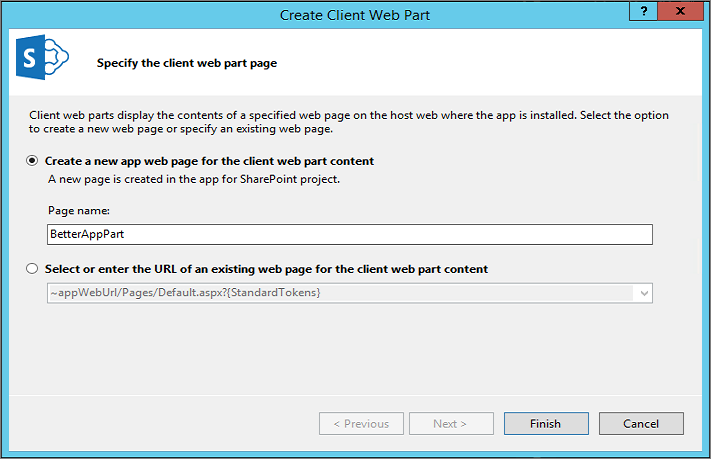
* 1. **Close the browser to end the debugging session and then return to Visual Studio.**

**Now you have created and tested a simple app part based on an HTML page. Next, you will create a more complicated app part with custom app part properties which is implemented with an ASPX file instead of a simple HTML file.**

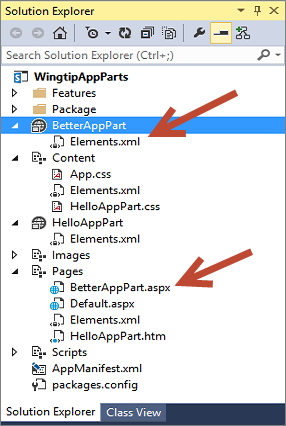
### Exercise 2: Creating an App Web with Custom Properties

In this exercise you will create and test an app part with custom properties.

1. Create a new app part named **BetterAppPart**,
   1. Right-click on the **WingtipAppParts** project and select the **Add New Item** command.
   2. In the **Add New Item** dialog, select the **Client Web Part (Host Web)** project item template (Located in the **Visual C# Items 🡪 Office/SharePoint** category) and give it a name of **BetterAppPart**. Click the **Add** button**, you should then see the Create Client Web Part dialog.**
   3. In the **Create Client Web Part** dialog, accept the default settings and click **Finish**.

. 

* 1. Once the Client Web Part has been added, inspect what files have been added to the project. You should see that Visual Studio created a folder named **BetterAppPart** for the project item which contains an elements.xml file which defines the Client Web Part. In addition, an aspx page named **BetterAppPart** has been added to the **Pages** folder.



1. Open the **elements.xml** file in the **BetterAppPart** and modify its content to look like this.

<?xml version="1.0" encoding="utf-8"?>

<Elements xmlns="http://schemas.microsoft.com/sharepoint/">

<ClientWebPart

Name="BetterAppPart"

Title="Better App Part"

Description="A really nice app part"

DefaultWidth="600"

DefaultHeight="200">

<Content

Type="html"

Src="~appWebUrl/Pages/BetterAppPart.aspx?{StandardTokens}" />

<Properties>

</Properties>

</ClientWebPart>

</Elements>

1. Save and close the **elements.xml** file.
2. Open **BetterAppPart.aspx** in Code View. Do not make any modifications to the **Page** directive, the **Register** directives or the **WebPartPages:AllowFraming** control at the top of the page. However, modify the HTML content below in the page to look like the code following listing. (i.e. this means you will remove all the <script> tags and associated script content (except for the jquery script tag) from the <head> section of the page in addition to adding content to the <body> section)

<%@ Page Language="C#" Inherits="Microsoft.SharePoint.WebPartPages.WebPartPage, BLAH BLAH BLAH %>

<%@ Register TagPrefix="SharePoint" Namespace="Microsoft.SharePoint.WebControls" BLAH BLAH BLAH %>

<%@ Register TagPrefix="Utilities" Namespace="Microsoft.SharePoint.Utilities" BLAH BLAH BLAH %>

<%@ Register TagPrefix="WebPartPages" Namespace="Microsoft.SharePoint.WebPartPages" BLAH BLAH BLAH %>

<WebPartPages:AllowFraming ID="AllowFraming" runat="server" />

<html>

<head>

<title></title>

<script type="text/javascript" src="../Scripts/jquery-1.9.1.min.js"></script>

</head>

<body>

<h4>Better App Part Content</h4>

<div id="results">default contents</div>

</body>

</html>

* 1. Save your changes to the **BetterAppPart.aspx** file. (Note: Keep this file open we will need it later)

1. Add some JavaScript code for the app part.
   1. In Windows Explorer, look inside the folder at **C:\Student\Modules\SharePointHostedApps\Lab\StarterFiles** and locate the file named **wingtip.utilities.js**. Add this file into the **WingtipAppParts** project in the **Scripts** folder. (Note: you can accomplish this by dragging the file from the **Starter Files** source folder in File Explorer into the **Scripts** destination folder in the Solution Explorer in Visual Studio)
   2. Inspect what's inside of **wingtip.utilities.js**. As you can see, it is a JavaScript Module named **Wingtip.Utilities** that is very similar to the one you created in the JavaScript programming lab.

'use strict';

var Wingtip = window.Wingtip || {};

Wingtip.Utilities = function () {

var getQueryStringParameter = function (param) {

var querystring = document.URL.split("?")[1];

if (querystring) {

var params = querystring.split("&");

for (var index = 0; (index < params.length) ; index++) {

var current = params[index].split("=");

if (param.toUpperCase() === current[0].toUpperCase()) {

return decodeURIComponent(current[1]);

}

}

}

}

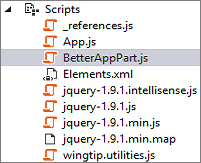
return {

getQueryStringParameter: getQueryStringParameter,

};

}();

* 1. Close **wingtip.utilities.js**.
  2. Add a new JavaScript file into the **Scripts** folder named **BetterAppPart.js**.
     1. Right-click on the **Scripts** folder in the **WingtipAppParts** project in **Solution Explorer** and select the **Add New Item** command.
  3. In the **Add New Item** dialog, select the **JavaScript File** template (Located in the **Visual C# Items 🡪 Web** category) and give it a name of **BetterAppPart.js**. Click the **Add** button**, you should then see the Create New JavaScript File dialog.**
  4. In the **Create New JavaScript File** dialog, accept the default settings and click **Finish**



* 1. **Add the following JavaScript code to BetterAppPart.js.**

****$(function () {****

****$("#results").text("My dynamic content");****

****$("body").css({****

****"border": "2px solid #CCC",****

****"padding": "8px"****

****});****

****$(":header").css({"border-bottom": "1px solid black"});****

****});****

* 1. **Save and close the BetterAppPart.js file.**

1. **Open BetterAppPart.aspx and add the following script links into the head section of the page:  
   (Note: you can do this quickly by dragging the JavaScript files from the Solution Explorer into the correct location in the BetterAppPart.aspx page)**
   1. **The jQuery library (verify that this is already there)  
      (Note: the version number on this library may differ from the code below as it is frequently updated)**
   2. **Wingtip.utilities.js**
   3. **BetterAppPart.js**

****<head>****

****<title></title>****

**<script type="text/javascript" src="../Scripts/jquery-1.9.1.min.js"></script>**

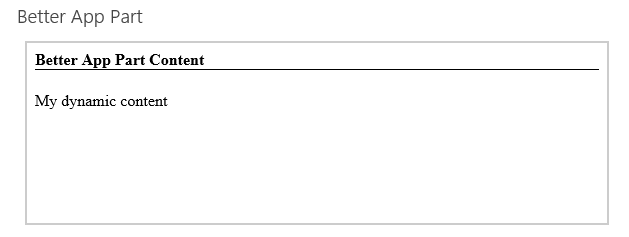
<script src="../Scripts/wingtip.utilities.js"></script>

<script src="../Scripts/BetterAppPart.js"></script>

****</head>****

* 1. **Save and close the BetterAppPart.aspx file.**

1. **Test your work by adding the BetterAppPart app part to a web part page in the host web.**
   1. **Press {F5} to begin a debugging session.**
   2. **When you see the app's start page, click the link to redirect to the home page of the host web.**
   3. **Use the Edit page menu from the Site Actions menu to move the page into Edit Mode.**
   4. **Once you are in Edit Mode, click the Add a Web Part link in the left web part zone to display the web part catalog.**
   5. **Locate and select the app part with a title of Better App Part in the Apps category folder. Click the Add button on the bottom right-hand side of the web part catalog to add the app part to the home page of the host web.**
   6. **In the Ribbon Bar Page Tab click Stop Editing. Now click the Browse Tab in the Ribbon to see your completed page with the app part.**
   7. **Once the app part is displayed, you should be able to verify that the JavaScript code executed property to add the message "My dynamic content" and to add a bottom border on the heading Better App Part Content.**

****

* 1. **Close the browser window to end the debugging session and return to Visual Studio.**

1. **Add two app part properties.**
   1. **Open the elements.xml file for the BetterAppPart app part. Add the two following property definitions.**

****<Properties>****

****<Property****

****Name="BackgroundColor"****

****WebDisplayName="Add Background Color"****

****Type="boolean"****

****DefaultValue="false"****

****WebCategory="Custom Wingtip Properties"****

****RequiresDesignerPermission="true" >****

****</Property>****

****<Property****

****Name="HeaderColor"****

****WebDisplayName="Header Color"****

****Type="enum"****

****DefaultValue="Black"****

****WebCategory="Custom Wingtip Properties"****

****RequiresDesignerPermission="true" >****

****<EnumItems>****

****<EnumItem WebDisplayName="Black" Value="Black"/>****

****<EnumItem WebDisplayName="Blue" Value="Blue"/>****

****<EnumItem WebDisplayName="Green" Value="Green"/>****

****</EnumItems>****

****</Property>****

****</Properties>****

* 1. **Inspect the Content element in elements.xml. Currently the Src attribute is defined as an URL which has a query string defined using only the dynamic token named {StandardTokens}.**

****<Content****

****Type="html"****

****Src="~appWebUrl/Pages/BetterAppPart.aspx?{StandardTokens}" />****

* 1. **Modify the query string in the elements.xml file as shown here to pass the custom property values to BetterAppPart.aspx.**

****BetterAppPart.aspx?BackgroundColor=\_BackgroundColor\_&amp;HeaderColor=\_HeaderColor\_&amp;{StandardTokens}****

* 1. **Save and close the elements.xml file.**
  2. **Return to BetterAppPart.js and add some code to read the two property values from the query string.**

****$(function () {****

****$("#results").text("My dynamic content");****

****$("body").css({****

****"border": "2px solid #CCC",****

****"padding": "8px"****

****});****

****$(":header").css({"border-bottom": "1px solid black"});****

****var BackgroundColor = Wingtip.Utilities.getQueryStringParameter("BackgroundColor");****

****if (BackgroundColor === "true") {****

****$("body").css({ "background-color": "Yellow" });****

****}****

****var HeaderColor = Wingtip.Utilities.getQueryStringParameter("HeaderColor");****

****if (HeaderColor) {****

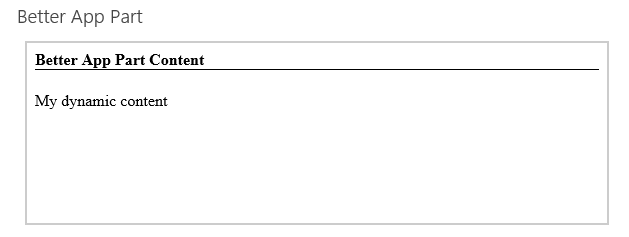
****$(":header").css({ "color": HeaderColor });****

****$(":header").css({ "border-bottom": "1px solid " + HeaderColor });****

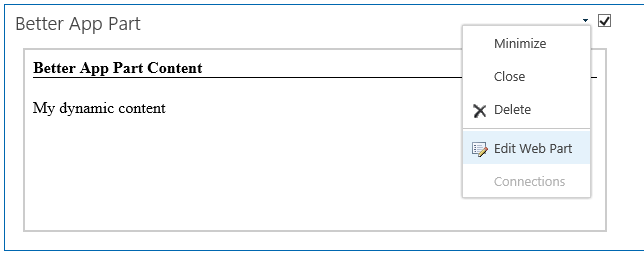
****}****

****});****

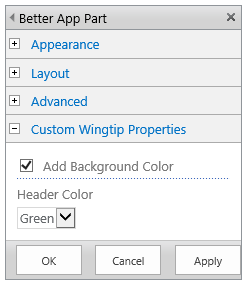
1. **Save and close the BetterAppPart.js file.**
2. **Test your work.**
   1. **Using the Visual Studio Build Menu select Rebuild WingtipAppParts to ensure the updated code will be deployed.**
   2. **Press {F5} to begin a debugging session.**
   3. **When you see the app's start page, click the link to redirect to the home page of the host web.**
   4. **Use the Edit page menu from the Site Actions menu to move the page into Edit Mode.**
   5. **Once you are in Edit Mode, click the Add a Web Part link in the left web part zone to display the web part catalog.**
   6. **Locate and select the app part with a title of Better App Part in the Apps category folder. Click the Add button on the bottom right-hand side of the web part catalog to add the app part to the home page of the host web.**
   7. **Once the app part is displayed, you should be able to verify that the JavaScript code executed property to add the message "My dynamic content" and to add a bottom border on the heading Better App Part Content.**



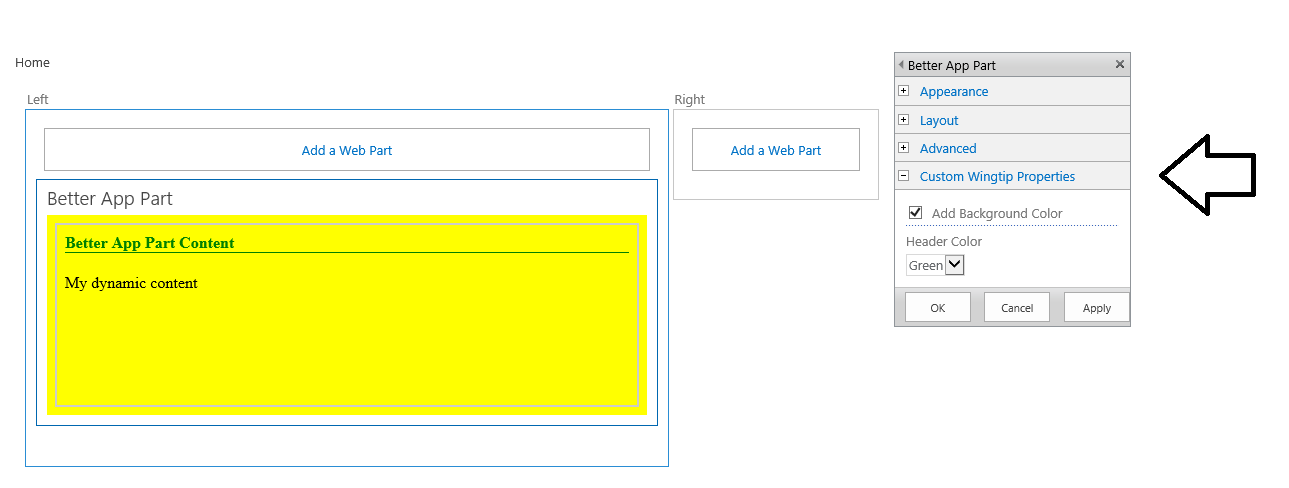
* 1. **As you are still in the Page Edit mode you can use the drop down app part menu in the top-right corner of the app part title bar to select the Edit Web Part menu. This will display the editor parts that make it possible for the user to modify app part properties.**

****

* 1. **In the editor part for the Better App Part, locate and expand the Custom Wingtip Properties section.**

****

* 1. **Enable the option to Add Background Color. Change the Header Color property to Green and then click the Apply button. You should see these changes affect the display the app part.**



* 1. **When you are done with your testing, close the browser window to end the debugging session.**

**You have now completed this lab where you have created and tested an app part with custom properties.**