Create and Deploy a SharePoint hosted App

in SharePoint 2013

1. Create a new Site Collection to test the App

a. Open **Central Administration** in SharePoint 2013 farm

b. Under '**Application Management**', select '**Create Site Collection**'. *(Make sure that only the appropriate Content database in which the site collection should be created in is kept online and the rest of the content databases if any, are offline.)*

c. In the Create Site Collection Page, give a ***Name*** of your choice and ***Path*****.**

d. In the template selection, choose **'Developer Site'**

e. Click **Create.**

f. Once the create process finishes, browse to the Site Collection.

g. Add '**sampleDomain\Appdeveloper'** as the **site collection administrator**.

2. Create a new Visual Studio Project

a. Logon to MSSP2013 SharePoint Server as '**sampleDomain\Appdeveloper'**

b. Start Visual Studio 2013 by using the ***Run as Administrator*** option.

c. On the menu bar, choose **File**, **New**, and **Project**.

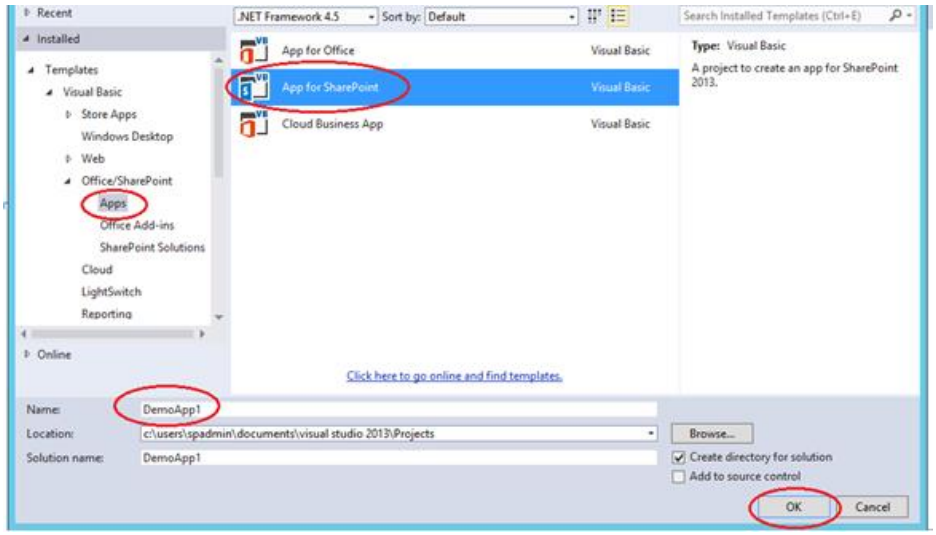
d. The New Project dialog box appears.

e. In the New Project dialog box, under **Visual Basic**, expand **Office/SharePoint**, and then choose the **Apps** category.

f. In the list of templates, choose the '**App for SharePoint'** template,

g. Rename the project as '**DemoApp1'**

h. Click **OK**.

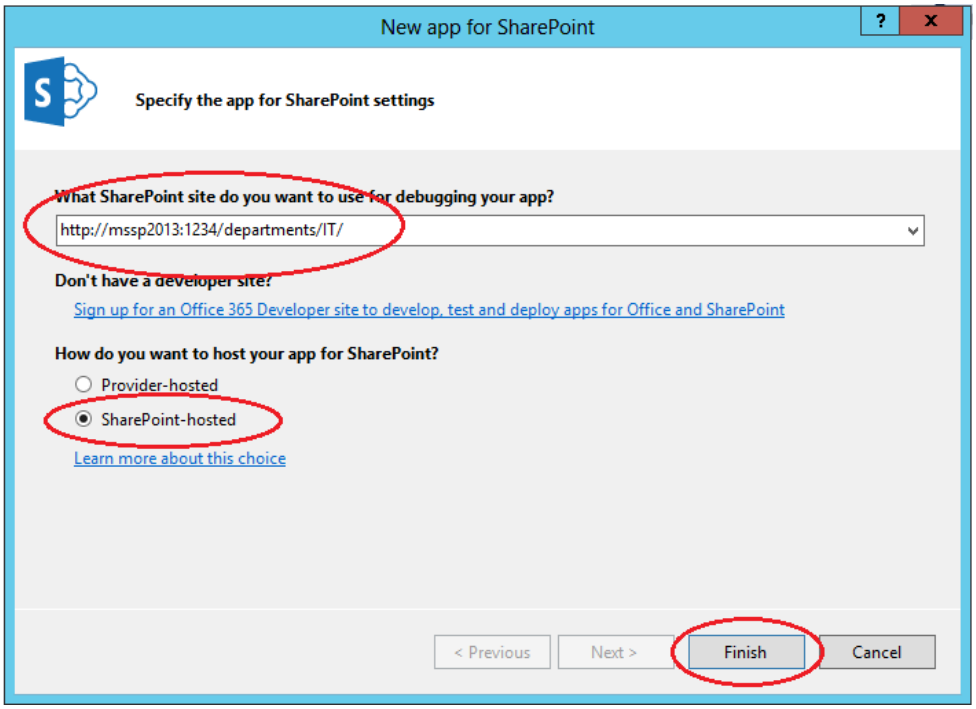


i. The SharePoint Customization Wizard appears.

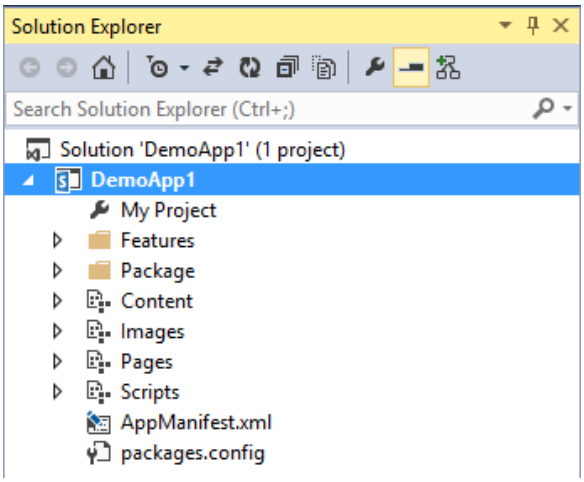
    i. By using this wizard, specify the site that you'll use to debug the project. (**URL for the site collection** we created in the previous section.)

j. In the 'How do you want to host your App?' Section, choose **'SharePoint Hosted'**

k. Choose the **Finish.**



l. Once the project is created, the solution explorer will list the following items:



m. Open the **scripts** folder, and view the **app.js** file.

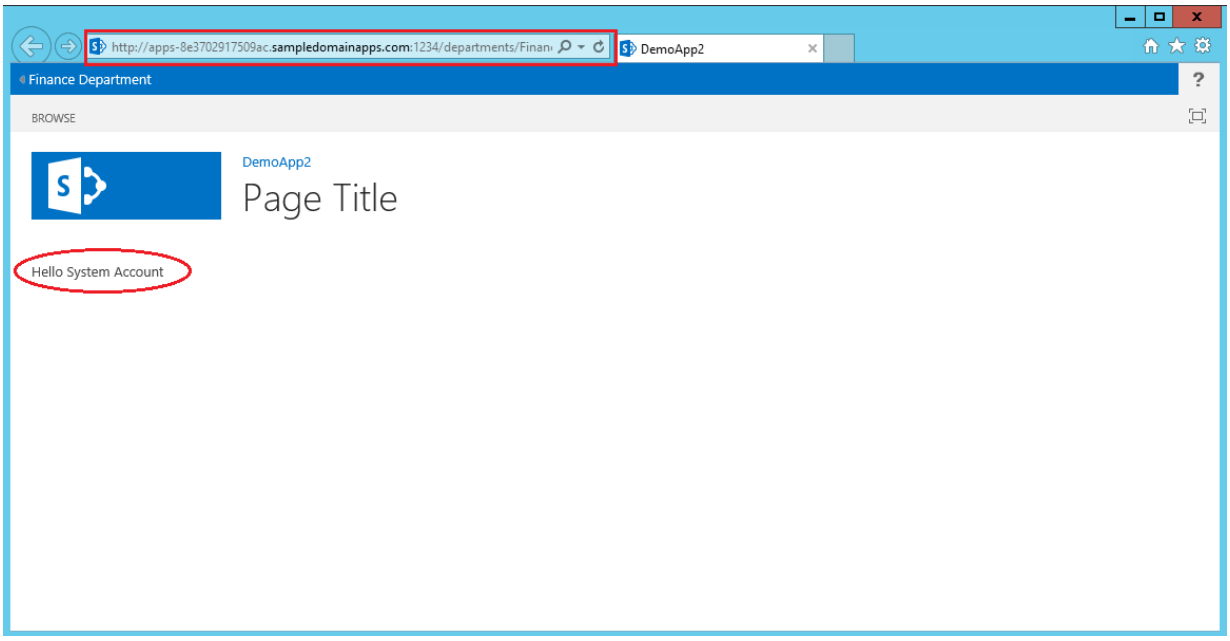
n. Visual Studio automatically inserts a sample JavaScript snippet to display '**Hello'** with the **logged in Username**.

3. Test the default Project

a. Build the project.

b. Start the project (**F5**)

c. The deploy for testing process will start and the browser page opens to show the default output as shown below:



d. Notice the **URL** of the App.

e. The App has opened in the **separate domain space** that we configured in a previous hands-on practice.

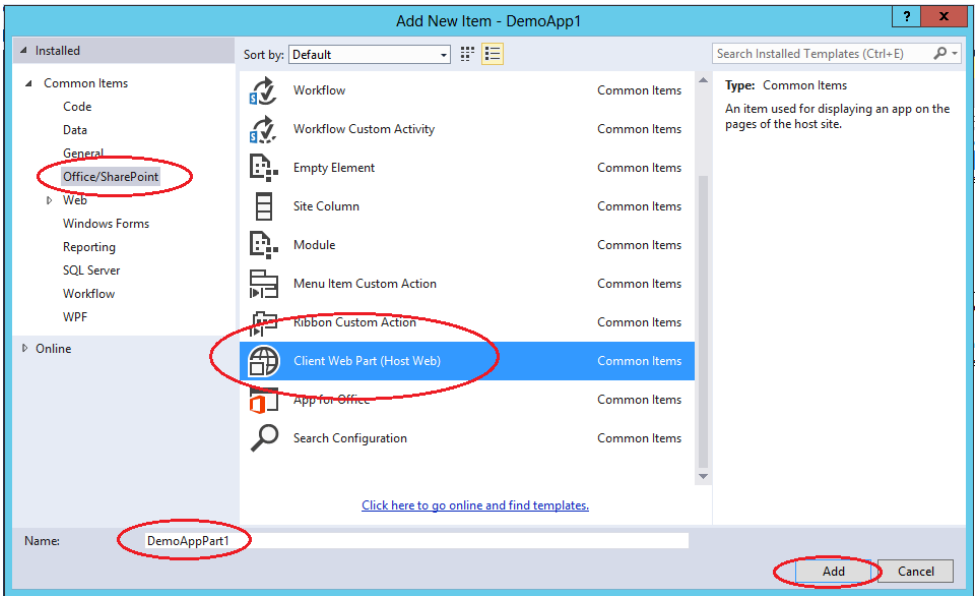
f. After this initial test for deploying SharePoint-hosted Apps, you can start building more functionality into the apps.

4. Add a Host Web (a.k.a *App Part or Client Part*) to the project

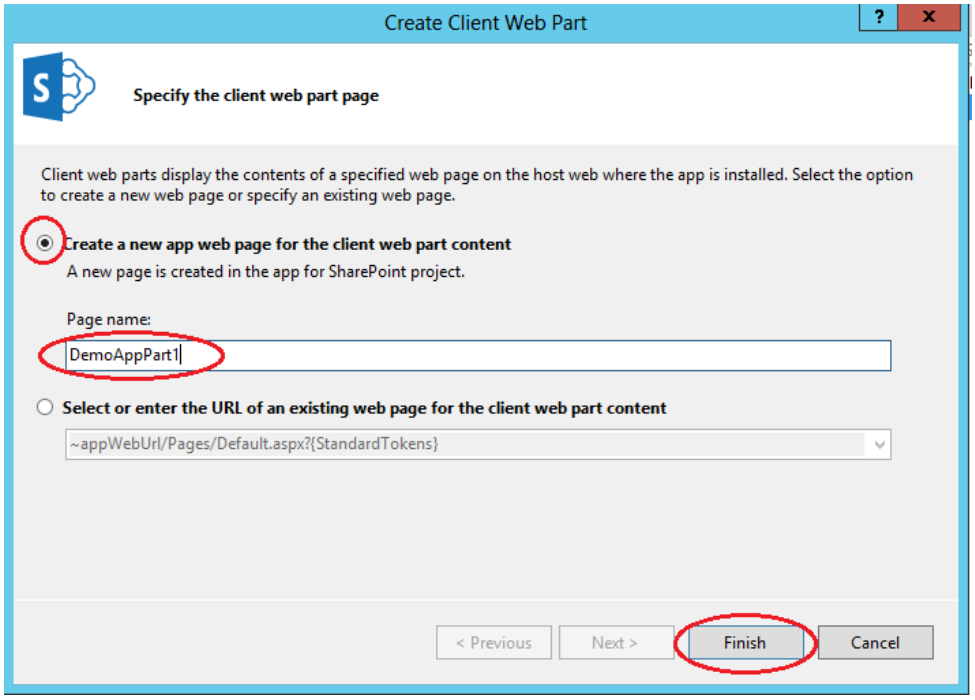
a. Right-click on the **Project Name** in solution explorer -> **Add New Item** - > **Client Web Part (Host Web).**

b. Change the name to **'DemoAppPart1'**

c. Click **'Add'.**

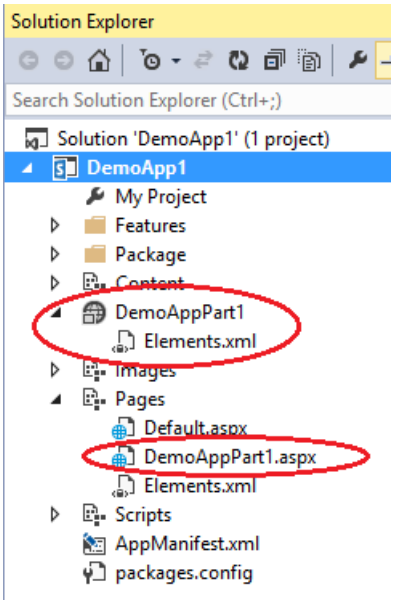


d. In the **'Create Client Web Part'** dialog box, choose the options as shown below**:**



e. Click **'Finish'**

f. A **new Client Web Part** and a **new .aspx Web Page** is created in your project as shown below:



a. You should notice that adding an App Part also add a corresponding **.aspx** file.

g. **Delete** the **default.aspx** file from your Project.

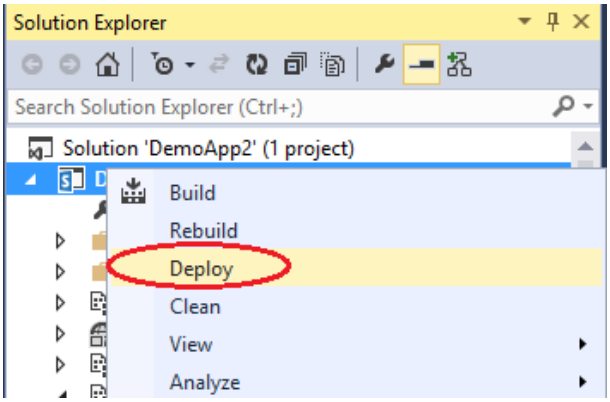
h. Open the '**DemoAppPart1.aspx'** file and add the following highlighted lines of code to it.

DemoAppPart1.aspx*

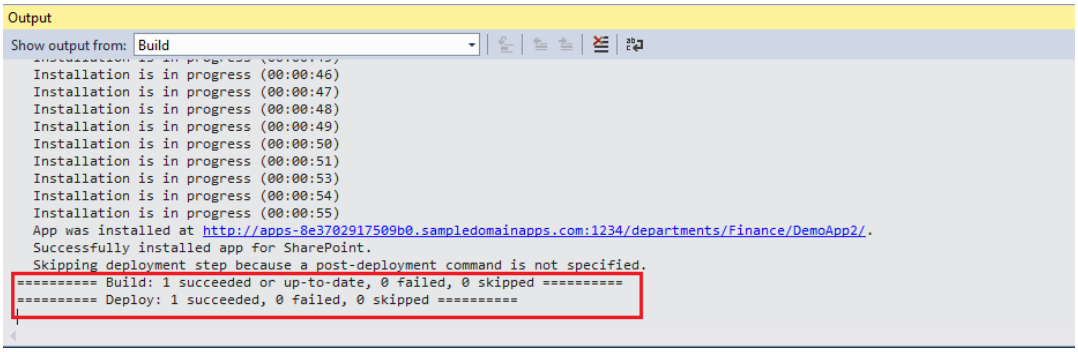
```
<%@ Page language="C#" Inherits="Microsoft.SharePoint.WebPartPages.WebPartPage, Microsoft.SharePoint, Version=15.0.0.0, Culture=neu
<%@ Register Tagprefix="SharePoint" Namespace="Microsoft.SharePoint.WebControls" Assembly="Microsoft.SharePoint, Version=15.0.0.0,
<%@ Register Tagprefix="Utilities" Namespace="Microsoft.SharePoint.Utilities" Assembly="Microsoft.SharePoint, Version=15.0.0.0, Cul
<%@ Register Tagprefix="WebPartPages" Namespace="Microsoft.SharePoint.WebPartPages" Assembly="Microsoft.SharePoint, Version=15.0.0.

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

<html>
<head>
    <title>My Demo App</title>

    <script type="text/javascript" src="../Scripts/jquery-1.9.1.min.js"></script>
    <script type="text/javascript" src="/_layouts/15/MicrosoftAjax.js"></script>
    <script type="text/javascript" src="/_layouts/15/sp.runtime.js"></script>
    <script type="text/javascript" src="/_layouts/15/sp.js"></script>

    <script type="text/javascript" src="../Scripts/App.js"></script>

    <script type="text/javascript">
        // Set the style of the client web part page to be consistent with the host web.
        (function () {
            'use strict';

            var hostUrl = '';
            if (document.URL.indexOf('?') != -1) {
                var params = document.URL.split('?')[1].split('&');
                for (var i = 0; i < params.length; i++) {
                    var p = decodeURIComponent(params[i]);
                    if (/^SPHostUrl=/i.test(p)) {
                        hostUrl = p.split('=')[1];
                        document.write('<link rel="stylesheet" href="' + hostUrl + '/_layouts/15/defaultcss.ashx" />');
                        break;
                    }
                }
            }
            if (hostUrl == '') {
                document.write('<link rel="stylesheet" href="/_layouts/15/1033/styles/themable/corev15.css" />');
            }
        })();
    </script>
</head>
<body>
  <div>
  <p id="message">
   <!-- The following content will be replaced with the user name when you run the app - see App.js -->
   initializing...
  </p>
 </div>

</body>
</html>
```

b. **Save** the project.

c. **Right-click** the project name and select '**deploy'**



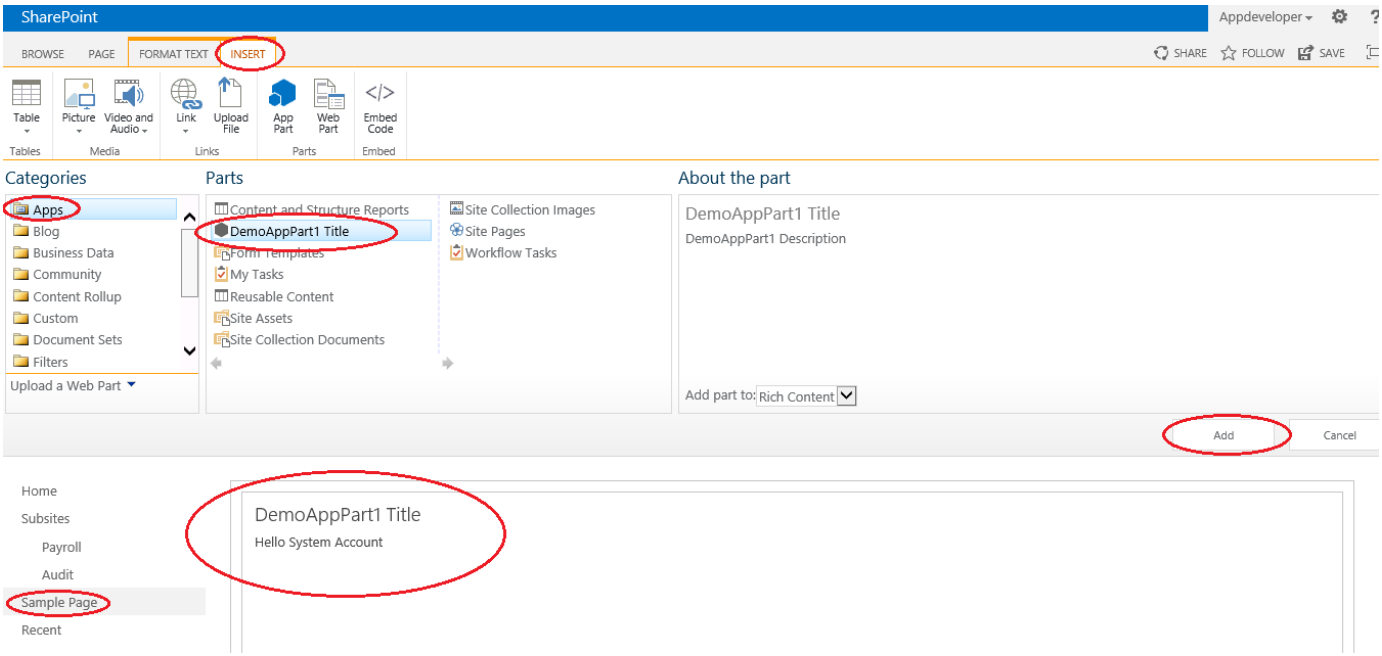
d. The **output window** will show the status of deployment.



e. Once the deployment is over, open the URL to the site collection used for the project creation.

f. Go to '**Site Content**', you should see the '**DemoApp2**' listed in the **App Gallery.**

g. Create a Test page in the site and add the App in the page from the App Gallery to test it as shown below:



h. You have now seen how by adding an App Part to the App, the App is now available in the app gallery to be added as stand-alone component into sites.

Note: You can add more App Parts and corresponding pages to the App Solution. Each App part and page will be listed separately in the INSERT-> App Gallery.
