App Deployment, Installation and Upgrade

Lab Time: 60 minutes

Lab Folder: C:\Student\Modules\PublishingApps\Labs

Lab Overview: Learning how to deploy publish and install apps correctly is a core skill that every SharePoint developer should know. In this lab you will create an app catalog, deploy apps, and update versions. Through the first few exercises, you will work with a SharePoint-hosted app named **Calculator App**. In the last exercise, you will also move through the steps required to deploy a provider-hosted app named **Wingtip Search App** in an on-premises environment which will require you to work directly with the Internet Information Services administration tool.

Exercise 1: Setup Lab Environment

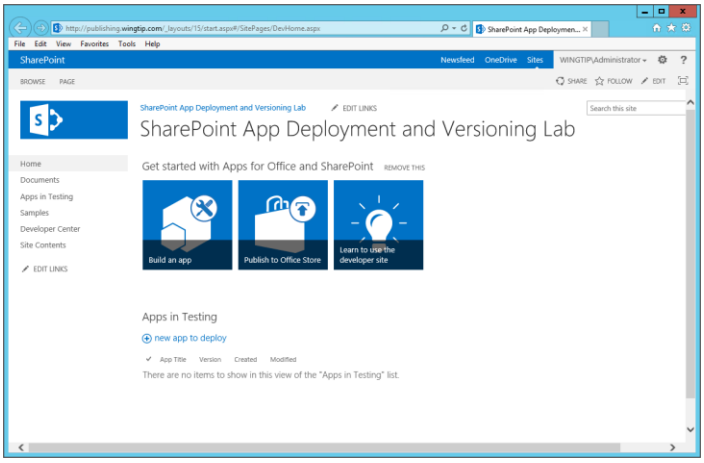
In this exercise you will setup your environment.

All exercises in this lab assume you will work in a new site collection, http://publishing.wingtip.com.

1. Setup a new site collection for this lab:
   a) Ensure you are logged into the **WingtipServer** server as **WINGTIP\Administrator**.
   b) Using Windows Explorer, navigate to the folder at the following path.

```
C:\Student\Modules\PublishingApps\Lab\
```

   c) Locate the PowerShell script named **SetupLab.ps1** and execute it by right-click it and selecting **Run with PowerShell**.
   d) When the script completes, it will launch a new browser and navigate to the lab site collection.

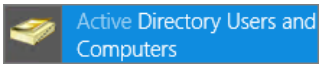


   e) Close the PowerShell console window.

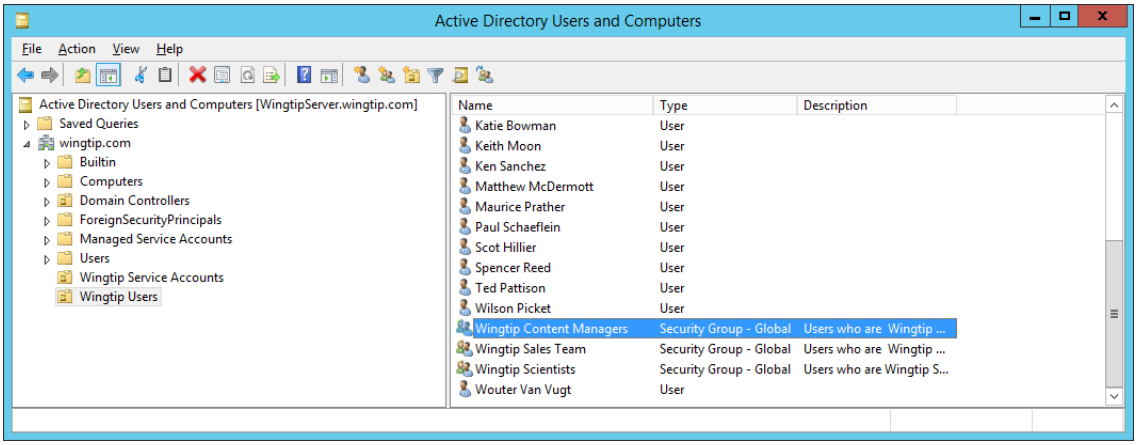
Exercise 2: Creating an App Catalog Site Collection

In this exercise you will create an app catalog to support app deployment and installation.

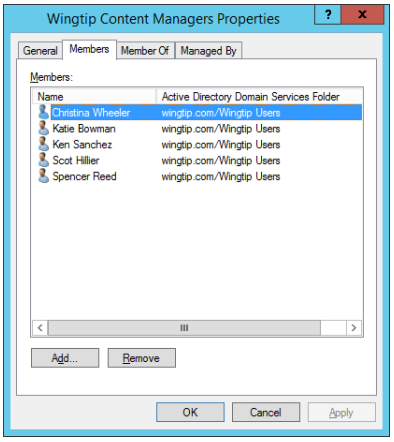
1. Review the user accounts and Active Directory groups created by the IT staff at Wingtip Toys.
   a) Open the **Active Directory Users and Computers** app.
      i) Press the **Windows** Keyboard Key.
      ii) Type **Active.**
      iii) Launch the **Active Directory Users and Computers** administrative application.



   b) Expand the **Wingtip Users** node and inspect the left-hand side of the window with the list of the user accounts and groups.



c) You should see that there is an Active Director group named **Wingtip Content Managers**. Double click this group to see the **Wingtip Content Managers Properties** dialog. You can see who's actually in this group using the **Members** tab.



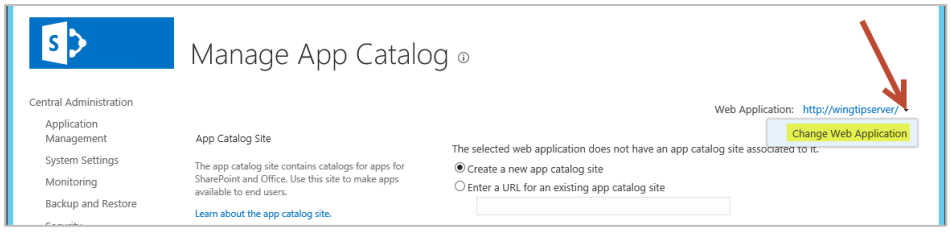
d) Click OK to close the **Wingtip Content Managers Properties** dialog.

If for any reason your student VM does not have an Active Directory group named **Wingtip Content Managers**, then create a new group using this name so you can continue on with this lab.

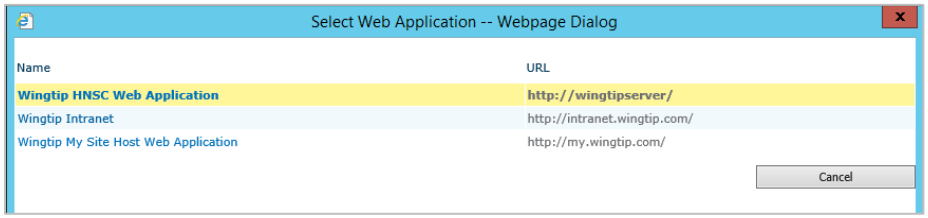
At this point you are done with the **Active Directory Users and Computers** administrative application and can close it if you'd like.

Creating the App Catalog Site Collection

2. Open Central Administration for the local SharePoint farm.
   a) Press the **Windows** Keyboard Key.
   b) Type Central.
   c) Launch the **SharePoint 2013 Central Administration** tool by clicking on the appropriate tile.
3. Create the App Catalog.
   a) On the home page of Central Administration, click **Apps → Manage App Catalog**.
   b) On the **Manage App Catalog** page, select the **Change Web Application** menu option.

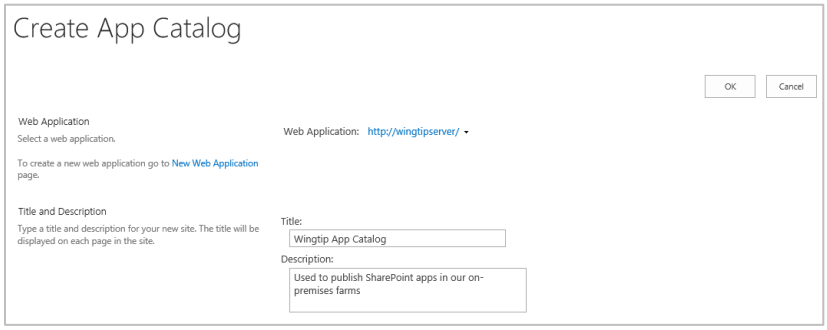


c) On the **Select Web Application – Webpage Dialog**, select the web application with the name of **Wingtip HNSC Web Application** and the URL of **http://wingtipserver**.



d) On the **Manage App Catalog** page, select the option to **Create a new app catalog site** and click **OK**.

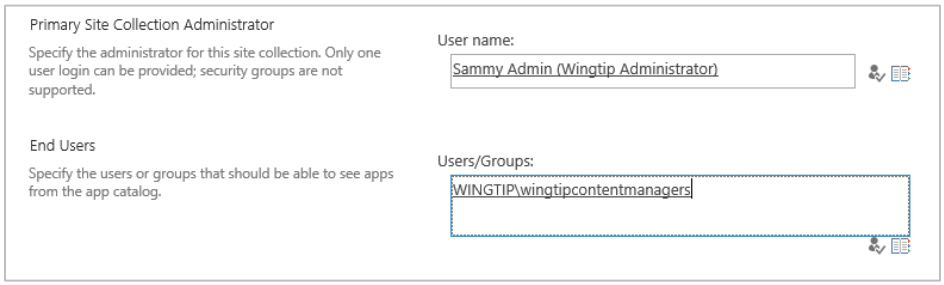
e) On the **Create App Catalog** page, enter a title of **Wingtip App Catalog** and brief description.



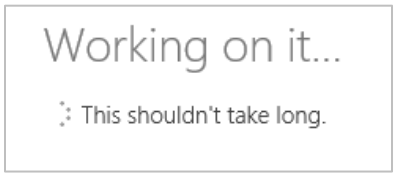
f) Move down in the **Create App Catalog** page to the **Web Site Address** section and configure the URL to a value of **http://wingtipserver/sites/AppCatalog**.

g) Move down in the **Create App Catalog** page to the **Primary Site Collection Administrator** section and assign the **User name** of **WINGTIP\Administrator**. This account may be resolved to a display name such as **Sammy Admin (Wingtip Administrator)**.

h) Move down in the **Create App Catalog** page to the **End Users** section and assign the Active Directory group named **Wingtip Content Managers**. This account may be resolved to a display name such as **WINGTIP\wingtipcontentmanagers**.



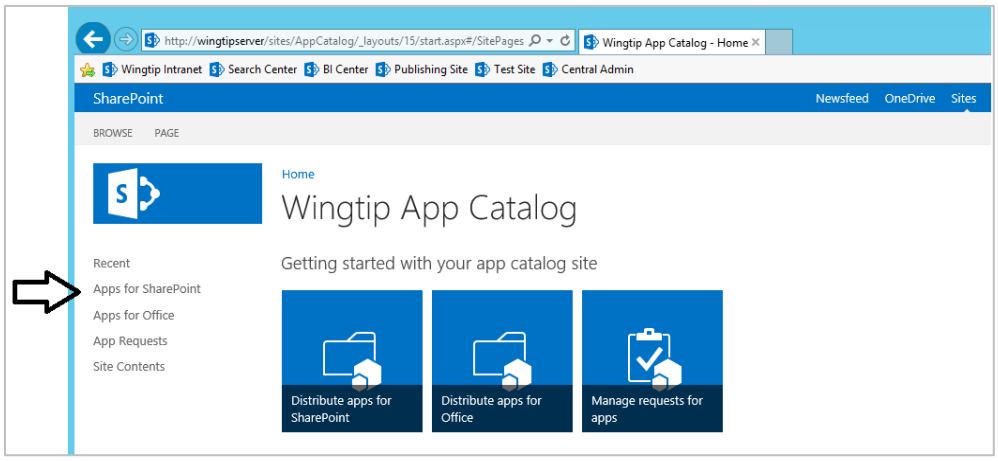
i) Move down to the bottom of the **Create App Catalog** page and click the **OK** button to begin the process of creating the new App Catalog site collection. SharePoint will display the notorious **Working on it…** animated GIF while it creates the new App Catalog site collection which can take anywhere from 30 seconds to 2-3 minutes.



j) Once SharePoint has finished creating the App Catalog site collection, you will be returned to the Manage App Catalog page which provides a link which you can click to navigate to the App Catalog site.

What if you wanted to create the App Catalog at a URL that was a top-level domain such as **http:\\appcatalog.wintip.com**? This is indeed possible with SharePoint 2013, but it cannot be accomplished using Central Administration. If you would like to create an App Catalog site collection as a top-level domain, you must use a PowerShell script which create it as a host-named site collection.

k) Click on the **http://wingtipserver/sites/AppCatalog** link to navigate to the new App Catalog site collection.



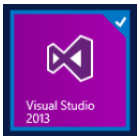
l) Note that the Quick Launch menu has a **Apps for SharePoint** link which you can use to navigate to the document library that is used to upload app packages in the app publishing process.

Now you are done configuring the local SharePoint farm for publishing and deploying apps. Now, you will work to publish and install several different SharePoint apps so you can get first-hand experience deploying SharePoint apps in an on-premises farm. However, keep in mind that an App Catalog is scoped at the Web Application level not the farm level. Therefore, the apps published in this App Catalog site collection can only be used by sites inside the same Web Application.

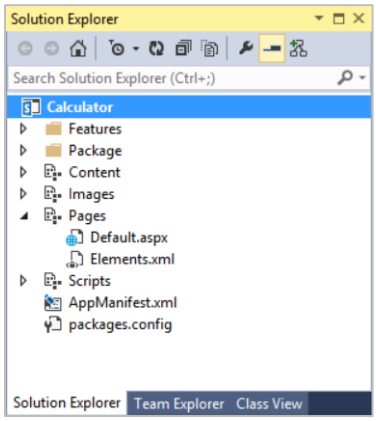
Exercise 3: Package and Deploy a SharePoint-Hosted App

In this exercise you will package an existing SharePoint-Hosted app and publish it to the Wingtip App Catalog site.

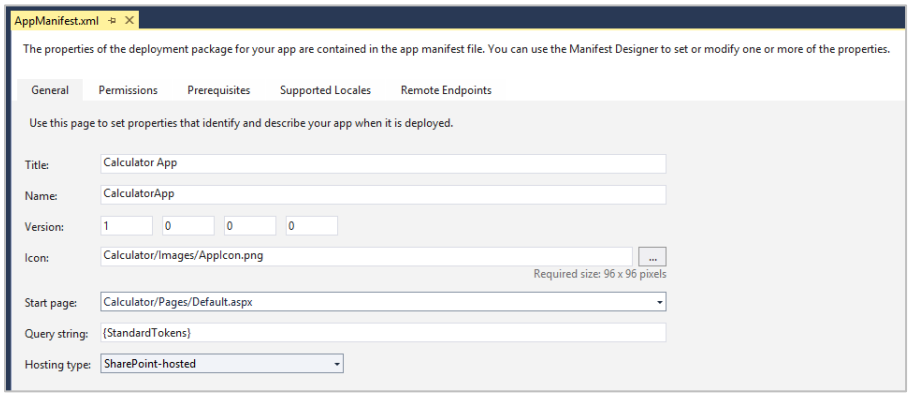
1. Launch **Visual Studio 2015** as administrator:
   a) **Windows** Keyboard Key → Right click on the Visual Studio 2015 tile and select **Run as administrator**.



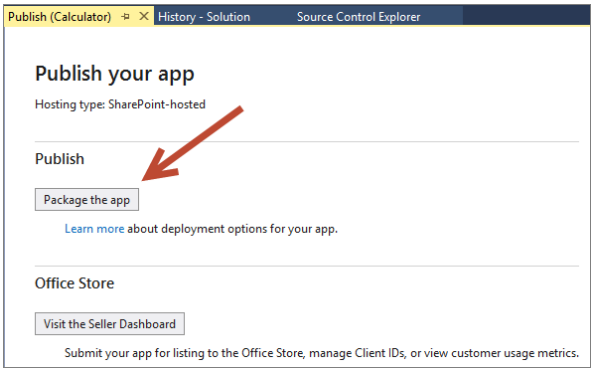
2. Open the starter solution in Visual Studio 2015:
   a) In Visual Studio, select **File → Open → Project/Solution**
   b) In the **Open Project** dialog:
      i) Browse to the **C:\Student\Modules\PublishingApps\Lab\StarterProjects\Calculator** folder
      ii) Double click on **Calculator.sln** to open the solution.
3. Examine the structure of the **Calculator** project. The project has the standard structure for a SharePoint-hosted app with a start page named **Default.aspx** in the **Pages** folder.



4. Inspect the App Manifest.
   a) Double-click on the **AppManifest.xml** to take a look at the app manifest settings.
   b) Review the current settings in the app manifest and take note that many of these settings are used by SharePoint when you publish, install and upgrade apps though the App Catalog.

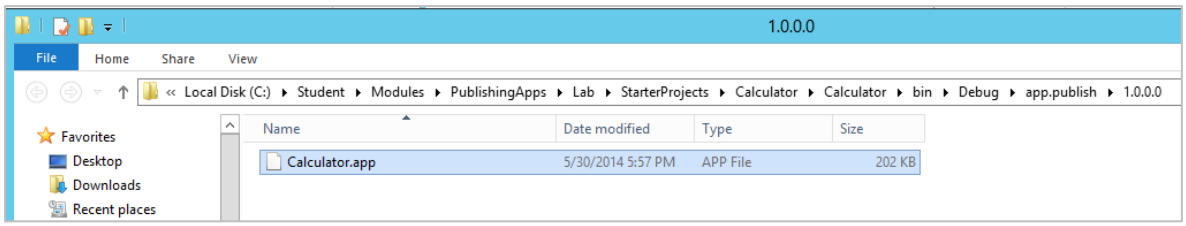


5. Package the App.
   a) In the **Solution Explorer**, right-click the **Calculator** project node.
   b) Select **Publish…** from the context menu.



c) In the Publishing Wizard, click **Package the app**.

d) When you click **Package the app**, Visual Studio builds the app package and then opens the Windows Explorer at the folder which contains the new app package.



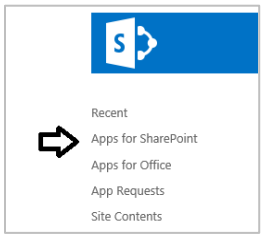
e) Keep Visual Studio opened as we will need this same solution in a later exercise.

6. Publish the App

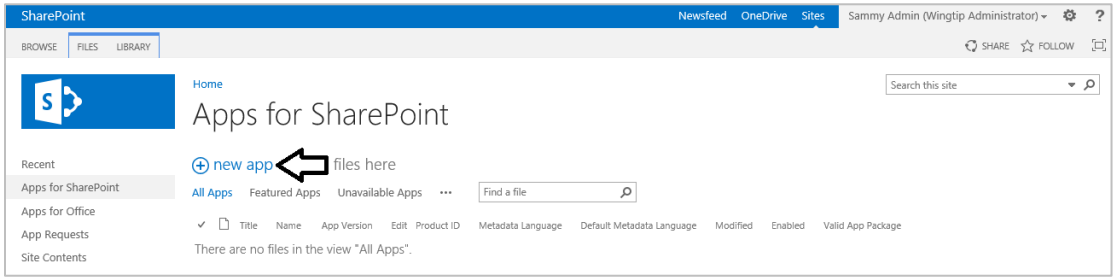
a) In the browser, navigate to the **Wingtip App Catalog** site at the following URL.

```
http://wingtipserver/sites/AppCatalog
```

b) Click the **Apps for SharePoint** link in the left navigation area.



c) On the page for the **Apps for SharePoint** library, you will see a **new app** link as shown in the following screenshot.

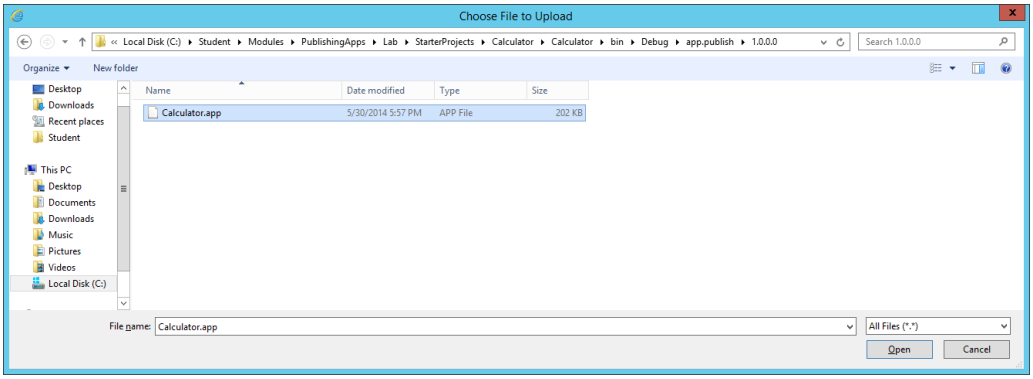


d) Upload the package:

i) Click the **new app** link to display the **Add a document** dialog.

ii) On the Add a document dialog, click **Browse** to display the **Choose File to Upload** dialog.

iii) Select the app package named **Calculator.app** and then click **Open** to close the **Choose File to Upload** dialog.

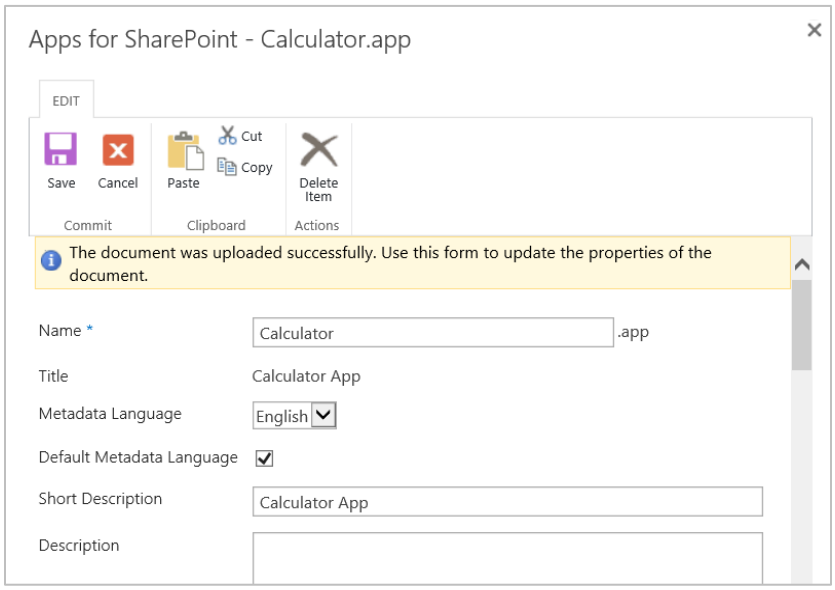


iv) Click **OK** to upload the app package and close the **Add a document** dialog.

e) At this point you should be prompted with a properties page for **Calculator.app**.

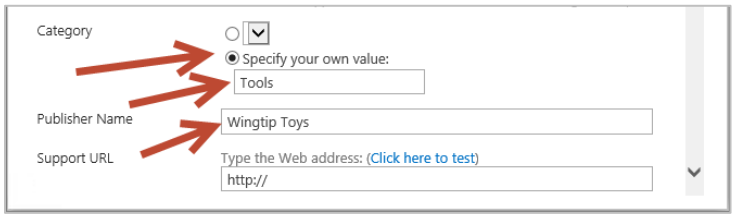
f) Accept the default value for **Name** which is just the file name of the app package.

g) Enter **Calculator App** for the **Short Description**.

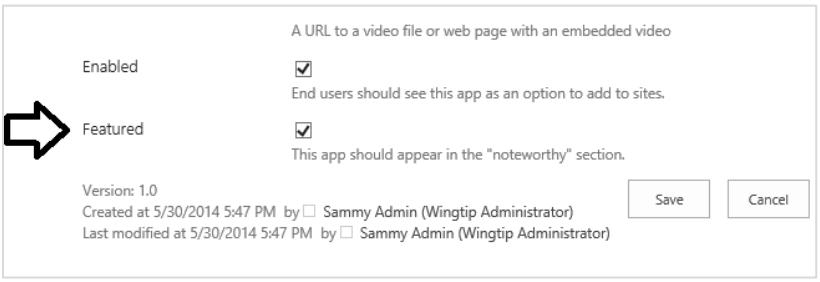


h) Enter **Tools** for the **Category**.

i) Enter **Wingtip Toys** for the **Publisher Name**.



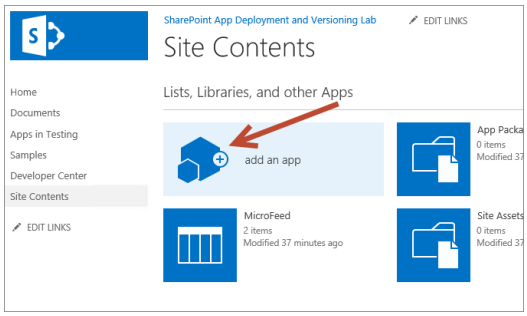
j) Scroll down to the bottom of the properties page for **Calculator.app** and select the **Featured** property by checking the checkbox.



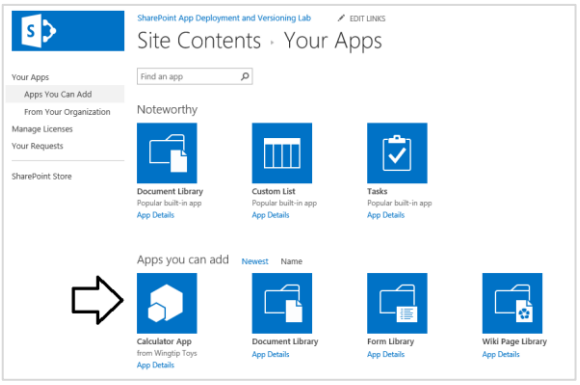
k) Click **Save** to save the metadata properties associated with the **Calculator** app.

Install the Calculator App at Site Scope

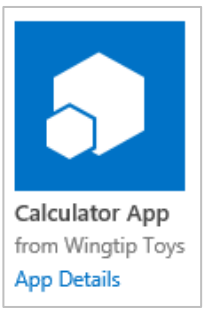
7. In Internet Explorer, **Navigate** to your site http://publishing.wingtip.com, where you can test install the app.
8. Click **Site Contents** in the Quick Launch menu**.**
9. Click **Add an App**.



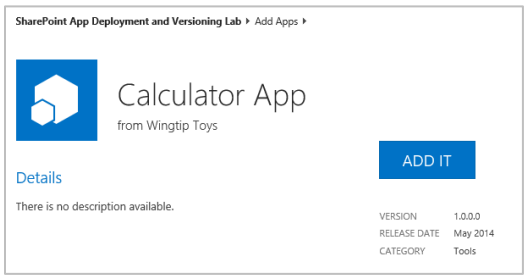
10. On the **Site Content – Your Apps** page, you should see that **Calculator App** listed as the first app in the **Apps you can add** section. The reason that the **Calculator** app is listed first in this section is because you configured it as a **Featured** app.



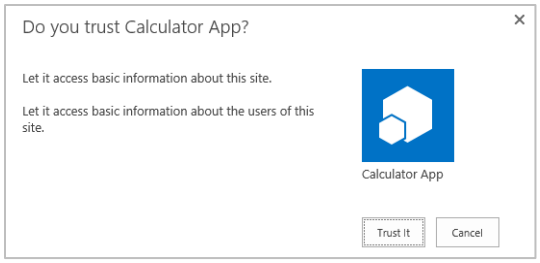
11. Click **App Details** under the Calculator App icon.



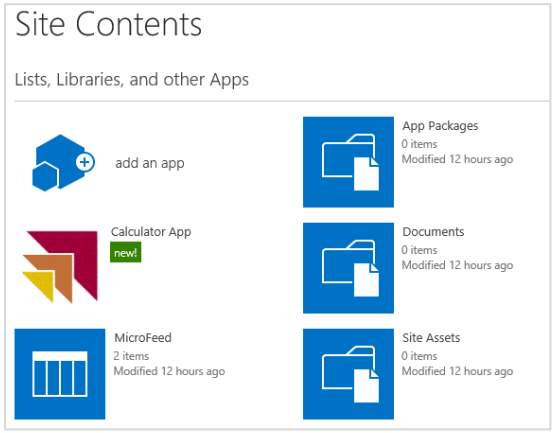
12. When you click **App Details**, you are redirected to the **Calculator App** page which list the app's details and provides a big blue **ADD IT** button.
13. Install the app.
    a) Click the **ADD IT** button to begin the app installation process.



    b) When prompted **Do you trust Calculator App**, click **Trust it**.

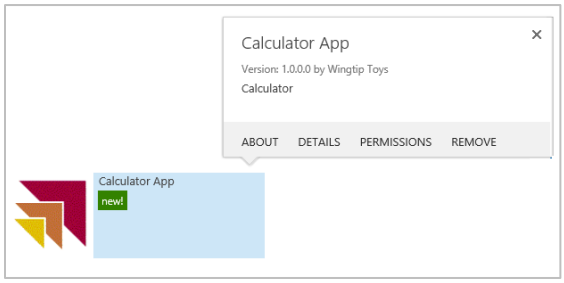


    c) At this point, you should see the app installing with a new tile on the Site Contents page. After a second or two the app will appear to be completely installed.

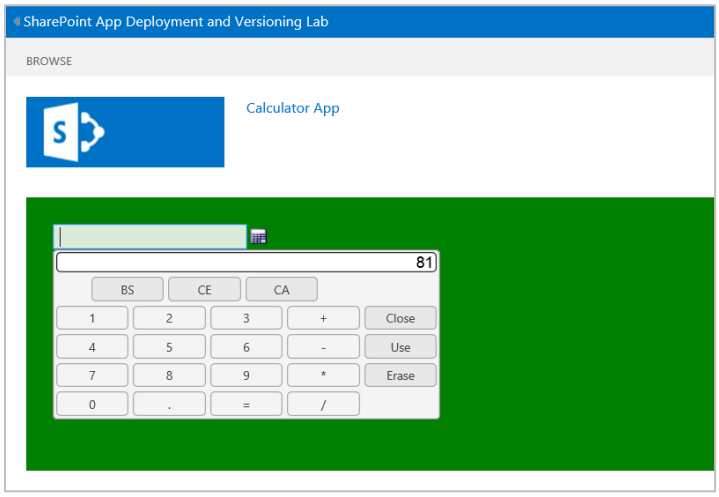


This is the point in the installation process where SharePoint has created an app web for the installed app instance.

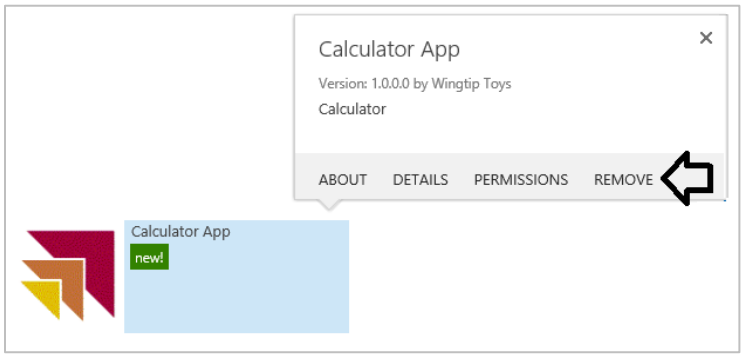
    d) Hover over the tile for the **Calculator App** and expand the flyout menu.



14. Give the **Calculator App** a quick test drive.
    a) Click on the **Calculator App** tile on the **Site Content** page to navigate to the **Calculator App** start page.
    b) Click the little calculator icon to display Calculator.



    c) Try and add two numbers together.
    d) OK, so this app really doesn't do too much. But now you know what this app does.
    e) Click the link at the top left-hand side of the screen to navigate back to the host web at **http://publishing.wingtip.com**.
15. Now that you have tested Calculator App, let's see how to **uninstall** it.
    a) Return to the **Site Contents** page.
    b) From the flyout menu associated with **Calculator App**, click **Remove**.



    c) When prompted, click **OK** to confirm the app removal.
16. You should be able to confirm that the app has been uninstalled by seeing that it no longer appears on the **Site Contents** page.

You have just gone through the basic steps of packaging, publishing and installing apps in a SharePoint 2013 on-premises environment. However, the previous app installation took place at site scope. Now think about a scenario in which you want the Calculator App to be available to man users across multiple site collections. While you could repeat the exact same site-scoped app installation process across many site collections, this type of approach can lead to maintenance issues down the road with respect to installation, upgrade and app removal. In the next exercise you will learn to install an app at tenancy scope which can offer many benefits with respect to managing the app lifecycle.

Exercise 4: Install a SharePoint Hosted App at the Tenancy Scope

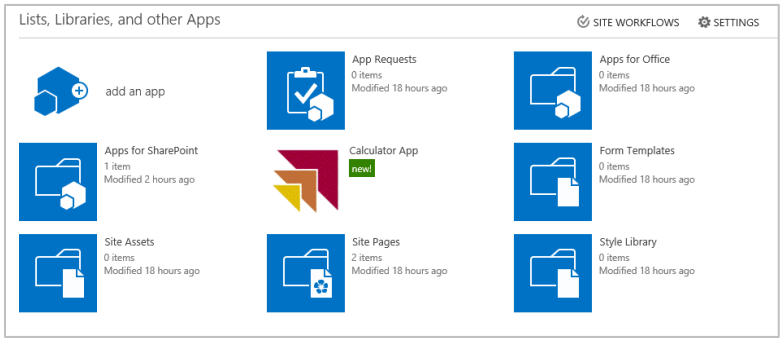
In this exercise, you will install **Calculator App** at the tenancy scope and configure it for use across all sites in a web application.

The only real trick to performing a tenancy-scoped installation is knowing where to install the app. Everything can be done from within the App Catalog site. After you have uploaded the app package to publish the app, you then install the app in the App Catalog site.

1. In the browser, navigate to the **Wingtip App Catalog** at **http://wingtipserver/sites/AppCatalog**.

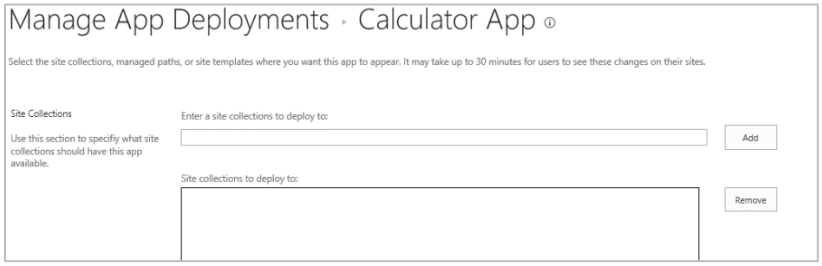
Remember earlier in this lab that you already published **Calculator App** to the App Catalog site. Next you will install the Calculator App into the same App Catalog site where it is published.

2. Install the **Calculator App** in the **Wingtip App Catalog site**.
   a) Click **Site Contents** link in the Quick Launch menu.
   b) Click **Add an App**.
   c) Underneath **Apps you can add** Click **Calculator App**.
   d) When prompted, click **Trust it**.
   e) The **Calculator App** should now be installed and appear on the **Site Contents** page of the App Catalog site.



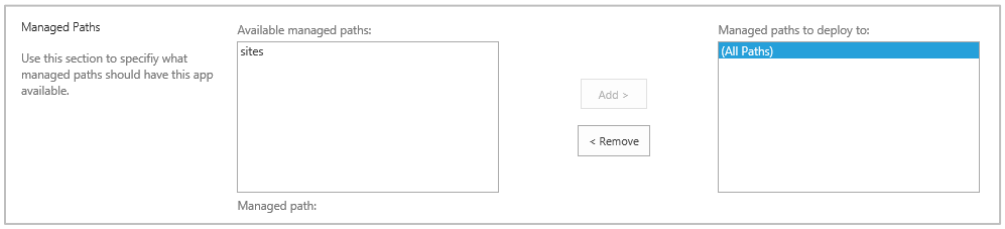
3. Configure the Deployments of Calculator App.
   a) From the **Calculator App** flyout menu, click **Deployment** to navigate to the **Manage App Deployments** page.

   b) As you can see, the **Manage App Deployments** page provides you with the ability to a URLs to site collections one by one. However, you will not use that approach in this lab.

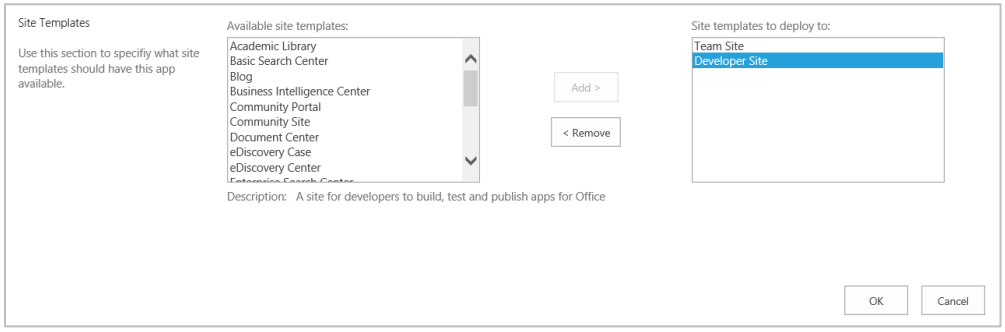


There are many different schemes you can to deploy the app to a set of sites within the current web application. For this lab, you will configure a catch-all deployment technique which will make Calculator App available to every site in the current web application.

   c) In the **Managed Paths** section, select **(All Paths)** to make the app available to all site.



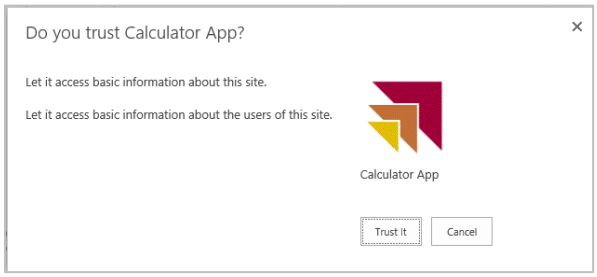
d) In the Site Templates section, select **Team Site** and **Developer Site**.



In general, you would not need to set anything in the **Site Templates** section given that the setting in the **Managed Paths** section that has been configured to target all sites in the current web application. We just want you to get a better sense of how you can configure target sites for the deployment.

e) At the bottom of the **Manage App Deployments**, click to OK button to push out your deployment configuration changes.

f) When prompted **Do you trust Calculator App**, click **Trust It**.



4. Take a second and think through the advantages of a tenancy-based installation you have just performed.
   a) Apps installed in this manner can only be uninstalled by an App Catalog administrator.
   b) Updates to the app become centrally managed.
   c) Updating the app can only be done by an App Catalog administrator.
   d) Updating a tenancy-scoped installation cascades all sites where the app is installed.

When you deploy apps with a tenancy-based installation, there is often a delay before the app actually appears in the target sites. The next step provides steps to make the apps appear sooner than they would in a normal scenario.

5. Run the PowerShell script named **PushAppCatalogChanges.ps1** to flush changes.
   a) Open the PowerShell ISE.
   b) Open the PowerShell script named **PushAppCatalogChanges.ps1** at the following path.

```
C:\Student\Modules\PublishingApps\Lab\PushAppCatalogChanges.ps1
```

   c) Review the PowerShell script code in **PushAppCatalogChanges.ps1**.

```
Add-PSSnapin Microsoft.SharePoint.PowerShell

# propagate update messahe as soon as possible
```

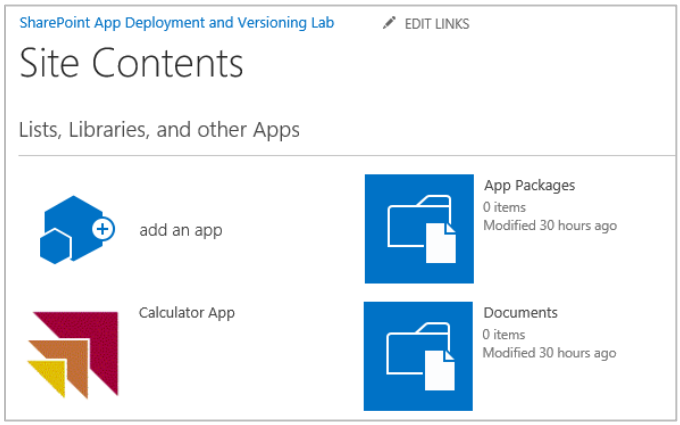
```
Set-SPAppStateUpdateInterval -AppStateSyncHours 0 -FastAppRevocationHours 0
Set-SPInternalAppStateUpdateInterval -AppStateSyncHours 0

# run timer jobs to propagate changes
$TimerJob1 =  Get-SPTimerJob | where { $_.DisplayName -eq 'App Installation Service' }
$TimerJob1 |  Start-SPTimerJob

$TimerJob2 =  Get-SPTimerJob | where { $_.DisplayName -eq 'App State Update' }
$TimerJob2 |  Start-SPTimerJob

$TimerJob3 =  Get-SPTimerJob | where { $_.DisplayName -eq 'Internal App State Update' }
$TimerJob3 |  Start-SPTimerJob
```

d) Execute the script.

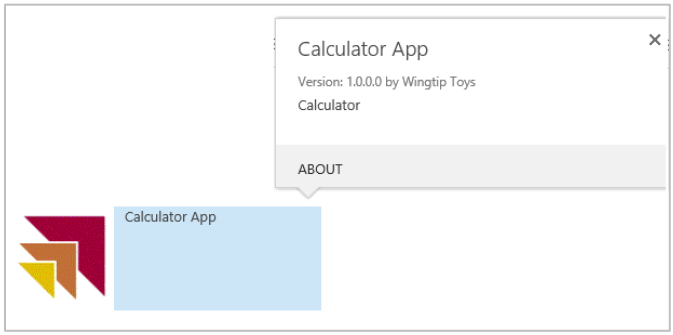
This script will cause the deployment changes of a tenancy-installed app to propagate much faster than a standard SharePoint farm which takes up to 24 hours to propagate all changes. However, there still might be a small delay (sometimes as many as 5-10 minutes) before you changes appear.

Now it's time to make sure things work and that the Calculator App is available in all sites within the target web application. As we have mentioned, if the Calculator App doesn't appear at first, you might have to wait a few more minutes before it appears.

6. Make sure **Calculator App** is available in the site at **http://publishing.wingtip.com**.
   a) In the browser, navigate to **http://publishing.wingtip.com**.
   b) Click on the **Site Contents** link in the Quick Launch menu.
   c) Verify you can see a tile for the **Calculator App** on the **Site Contents** page.



   d) If you expand the flyout menu for **Calculator app** you can see that that it only provides the **About** link. There is no option for the user to uninstall or configure the app in any way. Any configuration needs to be done back in the App Catalog site.



   e) Give the app a test drive by clicking on the **Calculator App** tile on the **Site Contents** page. You should see that the app appears just as before in that you can use the calculator.

7. Navigate to each of the following two site collections and verify the Calculator App is available.
   a) http://dev.wingtip.com
   b) http://security.wingtip.com

The Calculator App should be available in both these site collections because they reside within the same web application as the App Catalog site collection. However, if you now look for **Calculator App** in the site collection at the URL of **http://intranet.wingtip.com**, you will not see it. That's because that site collection and all the other site collections which have a root URL of **intranet.wingtip.com** reside in a different web application.

Exercise 5: Deploy an App Update

In this exercise, you will learn what is involved with updating an existing App by making a small update to the start page of an app and deploying the new version.

1. Return to Visual Studio 2015.
2. Open the Calculator project in Visual Studio if it is not still open.
   a) In Visual Studio, select **File → Open → Project/Solution**
   b) In the **Open Project** dialog:
      i) Browse to the **C:\Student\Modules\PublishingApps\Lab\StarterProjects\Calculator** folder
      ii) Double click on **Calculator.sln** to open the solution.

In this lab you will make a minor change to the app to demonstrate how to deploy an update. The change you will make is to change the behavior of the input control on the start page to be read-only.

3. Modify the App
   a) In the Solution Explorer, expand the **Pages** nodes.
   b) Double-click **Default.aspx**.
   c) Edit the HTML in the **PlaceHolderMain** content placeholder by replacing all of the content in the placeholder with the following code:

```
<p>Calculated Value: <input type="text" readonly="readonly" id="basicCalculator"/></p>
```

   d) Double-click **elements.xml**.
   e) Modify the **File** element to replace the app home page with the Default.aspx page by including a **ReplaceContent** attribute as shown in the following code:

```
<File Path="Pages\Default.aspx" Url="Pages/Default.aspx" ReplaceContent="TRUE" />
```

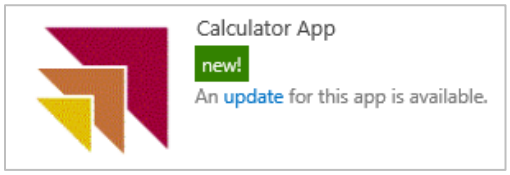
   f) In the Solution Explorer, double-click **AppManifest.xml**.
   g) Modify the **Version** attribute to be **2.0.0.0**.
   h) **Save** all changes to the project.
4. Modify the Feature manifest
   a) In the Solution Explorer, expand the **Features** folder.
   b) Double-click **Feature1**.
   c) Click the **Manifest** tab in the designer.
   d) Expand the **Edit Options** at the bottom of the designer.
   e) In the text block, replace all text with the following XML, which will be merged with the existing feature manifest:

```
<Feature xmlns="http://schemas.microsoft.com/sharepoint/">
   <UpgradeActions>
      <VersionRange BeginVersion="1.0.0.0" EndVersion="2.0.0.0">
         <ApplyElementManifests>
                <ElementManifest Location="Pages\Elements.xml" />
         </ApplyElementManifests>
      </VersionRange>
   </UpgradeActions>
</Feature>
```

5. Package the App
   a) In the **Solution Explorer**, right-click the **Calculator** project node.
   b) Select **Publish…** from the context menu.
   c) In the Publishing Wizard, click **Package the app**.

d) When finished, the File Explorer will open to the directory where the 2.0.0.0 version of the Calculator.app has been packaged.

6. Publish the App
   a) Navigate to the App Catalog site at http://wingtipserver/sites/AppCatalog.
   b) Click the **Apps for SharePoint** link in the Left Navigation area.
   c) Upload the version 2.0.0.0 package:
      i) Click **New App**.
      ii) Click **Browse** and locate the version 2.0.0.0 **Calculator.app** package you created.
      iii) Click **OK**.
   d) In the Properties dialog, click **Save** to keep the same properties as the previous version.
7. In a standard SharePoint 2013 farm, upgrade notifications might take as long as 24 hours to be pushed out. Here's an examples of what a notification looks like.



8. Upgrade the App
   a) In the **Wingtip App Catalog,** click the **Site Contents link** from the Left Navigation area.
   b) Using the flyout associated with the Calculator app, click **About**.
   c) On the **About** page, click **Get It** to upgrade version of the app which is **2.0.0.0**.
   d) When prompted, click **Trust It**.
   e) Wait for the upgrade to complete.
   f) Use the flyout menu of **Calculator App** to verify the version number is now **2.0.0.0**.

9. Test the upgraded version of **Calculator App** from the App Catalog site.
   a) **Launch** the **Calculator App**.
   b) Verify that the upgraded home page is visible.

In this exercise, we saw the process to use to update a SharePoint Hosted App that has been published by Tennant Administrators previously. Now that we're feeling pretty good about SharePoint Hosted App publishing… let's consider one additional possibility…

Exercise 6: Deploy a Provider-Hosted App in an On-premises Environment

In this exercise, you will deploy a provider-hosted app named Wingtip Search App to its own hosting infrastructure based on a new IIS website. First, you will create a new IIS website using a PowerShell script and then configure it to support Windows authentication and SSL using the Internet Information Services administration tool. After that you will work through the steps of deploying the ASP.NET application which provides the implementation of the remote web. Finally, you will build the app package for the Wingtip Sales App publish it to the App Catalog so you can test it and make sure it runs completely outside the Visual Studio environment.

1. Execute the PowerShell script named **StartFullCrawl.ps1** to crawl the local SharePoint Farm. This is required for later when the **Wingtip Search App** will need to search through and find content from within the local farm.
   a) Using Windows Explorer, locate the PowerShell script at the following path.

```
C:\Student\Modules\PublishingApps\Lab\StartFullCrawl.ps1
```

There is no need to wait for the **StartFullCrawl.ps1** script to complete before moving on to the next step.

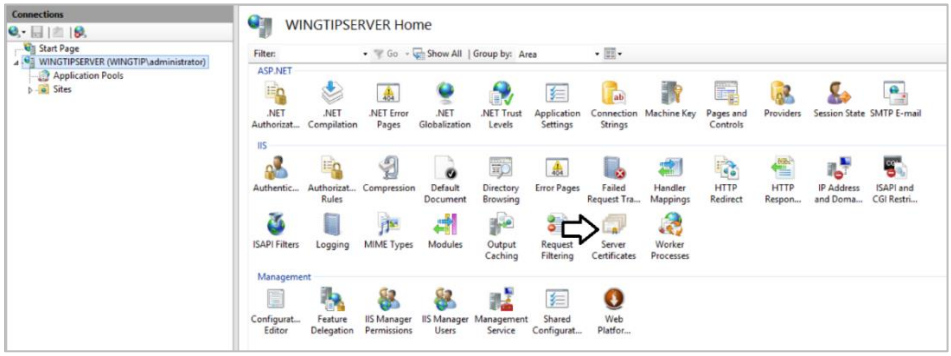
Create an IIS Website to Host the Remote Web

2. Examine the basic steps that lie ahead.
   a) Create an IIS web site with a root URL of **http://searchapp.wingtip.com**.
   b) Configure the IIS website to support Windows authentication.
   c) Configure the IIS website to support SSL through a URL of **https://searchapp.wingtip.com**.

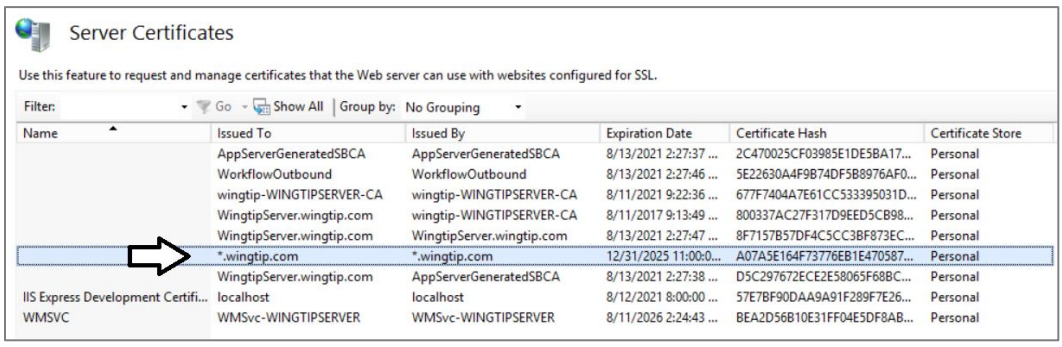
Microsoft strongly recommends using SSL whenever deploying a remote web. In fact, Visual Studio 2015 and the Office 365 environment will not let you configure a remote web URL that doesn't begin with **https**.

3. Launch the administrative tool for **Internet Information Services (IIS)**..
   a) Press the **Windows** key to display the Windows Start page.
   b) Locate and click the tile to launch the administrative tool for **Internet Information Services (IIS)**.

4. Find the wildcard SSL certificate for ***.wingtip.com**.
   a) On the left-hand side of the IIS administrative tool window, select the top-level **WINGTIPSERVER** node. With this node selected, locate and click the **Server Certificates** icon and click it to display the **Server Certificates** page.



   b) On the **Server Certificates** page, verify that you can see the SSL certificate with the **Issued To** value of ***.wingtip.com**.

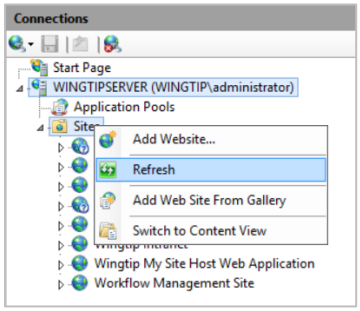


   c) Leave the IIS administrator tool open because you will be returning soon.

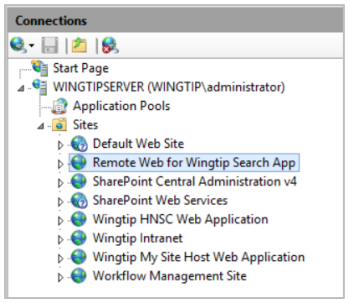
5. Execute the PowerShell script named **Create_IIS_WebSite_For_RemoteWeb.ps1** to create the IIS website.
   a) Using Windows Explorer, locate the PowerShell script at the following path.

   ```
   C:\Student\Modules\PublishingApps\Lab\Create_IIS_WebSite_For_RemoteWeb.ps1
   ```

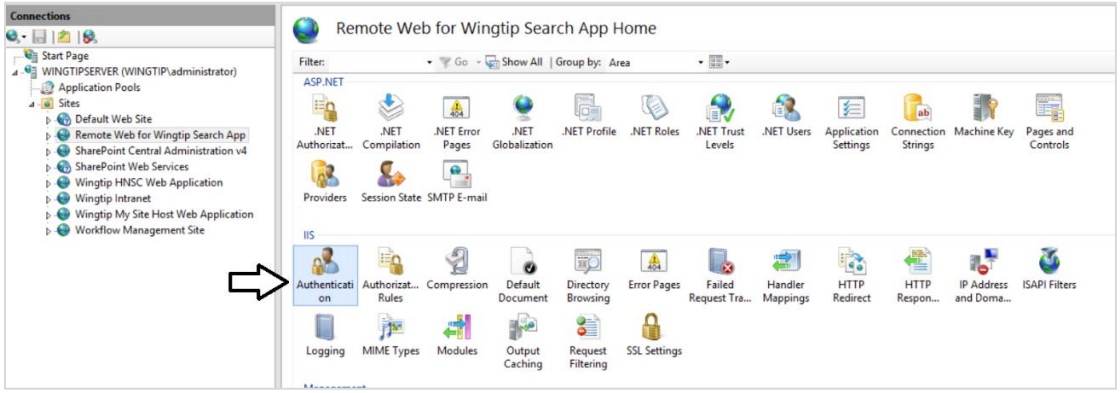
   b) Right-click **Create_IIS_WebSite_For_RemoteWeb.ps1** and click **Edit** to open the script in the PowerShell ISE.
   c) Execute **Create_IIS_WebSite_For_RemoteWeb.ps1** using the PowerShell ISE.
   d) Once the script has executed, return to the IIS administrative tool.
6. Verify that the new IIS website has been created.
   a) In the tree view on the left-hand side of the IIS administrative tool, right-click on the **Sites** node and select **Refresh**.



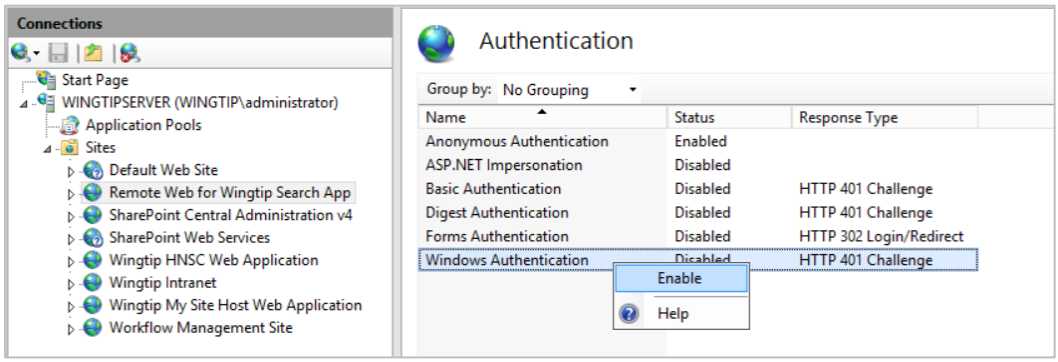
   b) Verify that you can see an IIS website named **Remote Web for Wingtip Search App**.



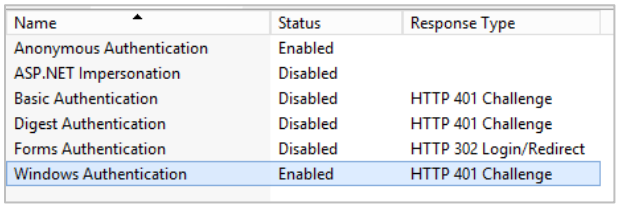
7. Configure the IIS website to support Windows authentication.
   a) Select the **Remote Web for Wingtip Search App** IIS website in the tree view control.
   b) Locate and double-click the **Authentication** tile in the IIS section as shown in the following screenshot.



   c) On the **Authentication** page, right-click the **Windows Authentication** row and then click **Enabled**.



d) Verify that Windows Authentication is the only type of authentication supported.



8. Configure the IIS website to support SSL.

a) Select the **Remote Web for Wingtip Search App** IIS website in the tree view control.

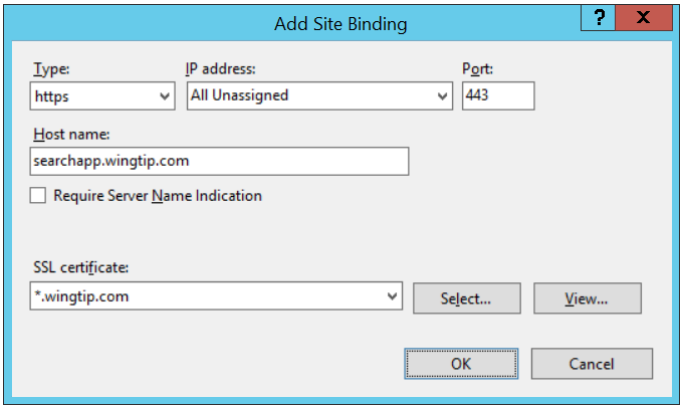
b) Locate and click the **Bindings links** on the far right in the Actions pane as shown in the following screenshot.

c) When the **Site Bindings** dialog appears, click to **Add** button to display the **Add Site Binding** dialog.

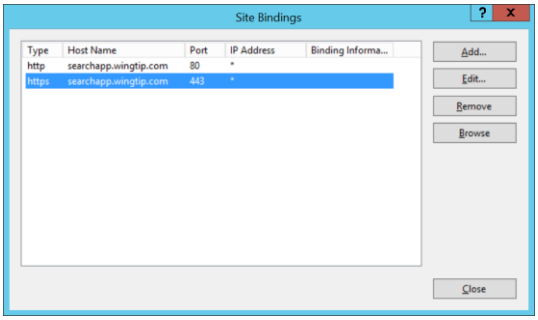
d) In the **Add Site Binding** dialog, set the type to **https** and the **Host name** to **searchapp.wingtip.com**.

e) Use the dropdown list for the **SSL Certificate** setting to select *.**wingtip.com**.

f) When you have filled out the **Add Site Binding** dialog like the following screenshot, click **OK** to save your changes.



g) The **Site Bindings** dialog should show the IIS website now supports SSL at **https://searchapp.wingtip.com**.



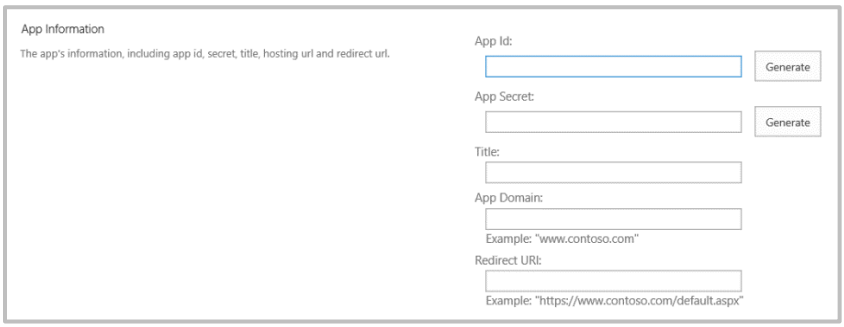
h) Click **Close** to close the **Site Bindings** dialog.

Register a Security App Principal for the Wingtip Search App

9. In the browser, navigate to the App Catalog site at **http://wingtipserver/sites/AppCatalog**.
10. Navigate to the SharePoint page for creating App Security Principals.
    a) Navigate to the following URL by directly typing it into the browser address bar.

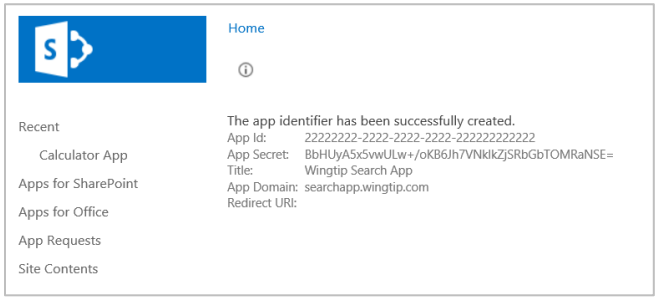
```
http://wingtipserver/sites/AppCatalog/_layouts/15/AppRegNew.aspx
```

    b) This is a special page added to SharePoint 2016 for creating App Security Principals.



11. Fill in the **AppRegNew.aspx** page using the following instructions.
    a) Enter an **App Id** of **22222222-2222-2222-2222-222222222222**.
        i) Note that the **Lab** folder has a text file named **2GUID2Btrue.txt** from which you can copy and paste this value.
    b) Click the **Generate** button to add a value for the App Secret.
    c) Add a **Title** of **Wingtip Search App**.
    d) Add an **App Domain** of **searchapp.wingtip.com**.
    e) Leave the **Redirect URL** empty.
    f) When your screen looks like the following screenshot, click the Create button to create the new App Security Principal.

g) After creating the new App Security Principal, SharePoint will display a confirmation page which shows the details.



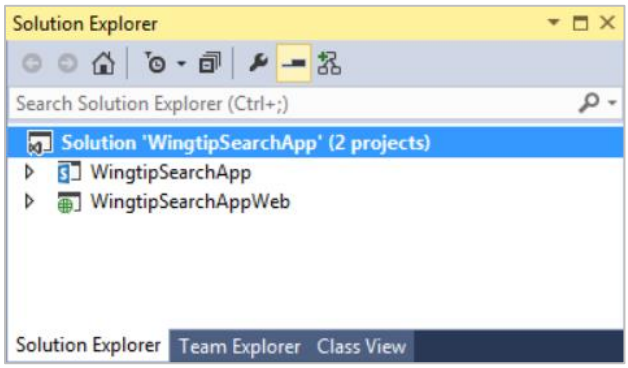
Now you are done creating the App Security Principal for the Wingtip Search App.

Deploy the Remote Web of the Wingtip Search App

12. Open the starter solution in Visual Studio 2015:
    a) Launch **Visual Studio 2015** as administrator:
        i) **Windows** Keyboard Key → Right click on the Visual Studio 2015 tile and select **Run as administrator**.
    b) In Visual Studio select **File → Open → Project/Solution**
    c) Locate and open the Visual Studio solution named **WingtipSearchApp.sln** at the following path.

```
C:\Student\Modules\PublishingApps\Lab\StarterProjects\WingtipSearchApp\WingtipSearchApp.sln
```

    d) As you can see, this is a typical solution for a provider-hosted app which contains two projects.
        i) **Open** the solution by double clicking on **WingtipSearchApp.sln**.

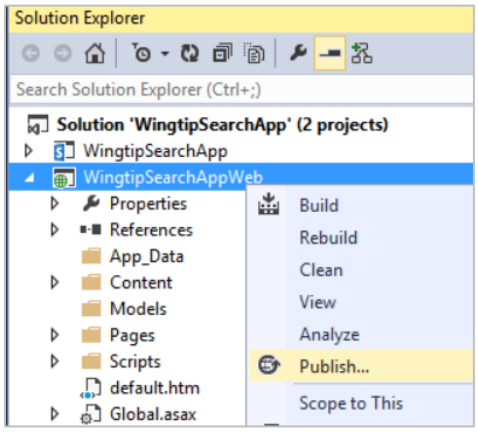


    e) In the Solution Explorer, expand the **WingtipSearchAppWeb** project node.
13. Double-click the **web.config** file to open it in an editor window.
    a) All the settings should already be set correctly. However, you should verify that the **appSettings** values that are used for S2S authentication have values as those shown in the following code listing.

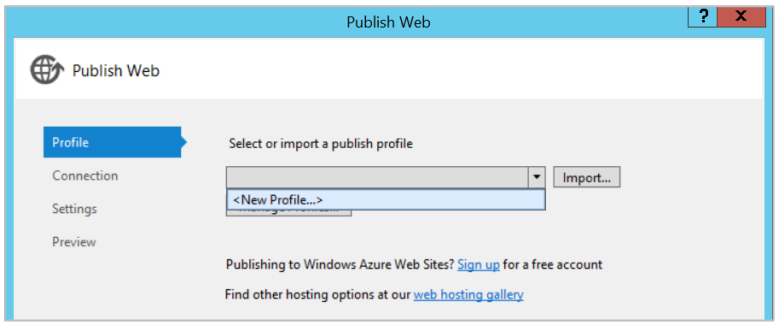
```
<appSettings>
  <add key="ClientId" value="22222222-2222-2222-2222-222222222222" />
  <add key="ClientSigningCertificatePath" value="C:\Certs\WingtipAppCertificate01.pfx" />
  <add key="ClientSigningCertificatePassword" value="Password1" />
  <add key="IssuerId" value="11111111-1111-1111-1111-111111111111" />
</appSettings>
```

Configure a Deployment Profile for the WingtipSearchAppWeb Project

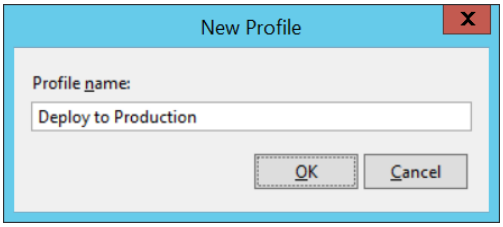
14. Create a new Deployment Profile for the **WingtipSearchAppWeb** project.
    a) Right-click the **WingtipSearchAppWeb** project node and select **Publish** to launch the **Publish Web** wizard.



b) The **Publish Web** wizard has four pages. Begin at the first page which is the **Profile** page and select **<New Profile…>**.



c) Create a nee profile named **Deploy to Production**.



d) Click **Next** to move to the **Connection** page.

e) Fill the **Connections** page out with the following information:

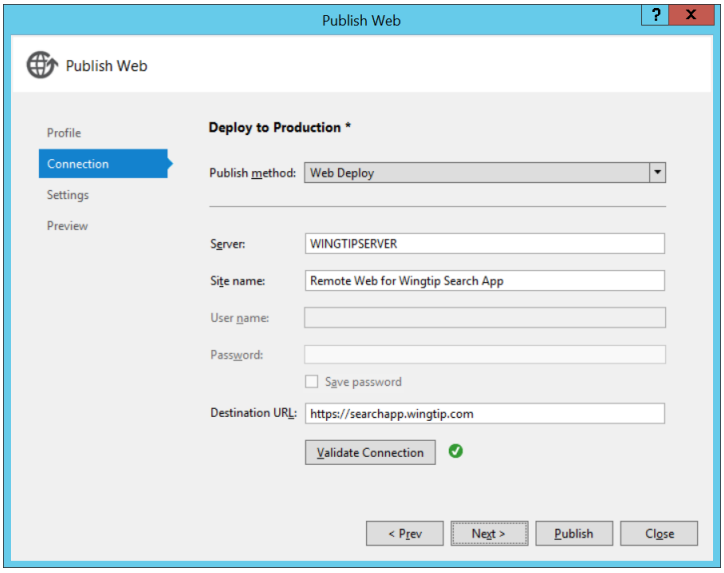
i) **Publish method**: Web Deploy

ii) **Server**: WINGTIPSERVER

iii) **Site name**: Remote Web for Wingtip Search App

iv) **Destination URL**: https://searchapp.wingtip.com

f) Click on the **Validate Connection** button to ensure the connection information is valid.



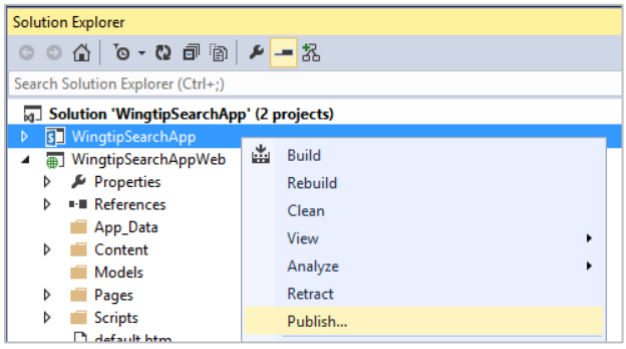
g) Click **Publish** to deploy the ASP.NET project into the IIS website at **https://searchapp.wingtip.com**.

If deployment is successful, you should be able to access a generic landing page at **https://searchapp.wingtip.com.**

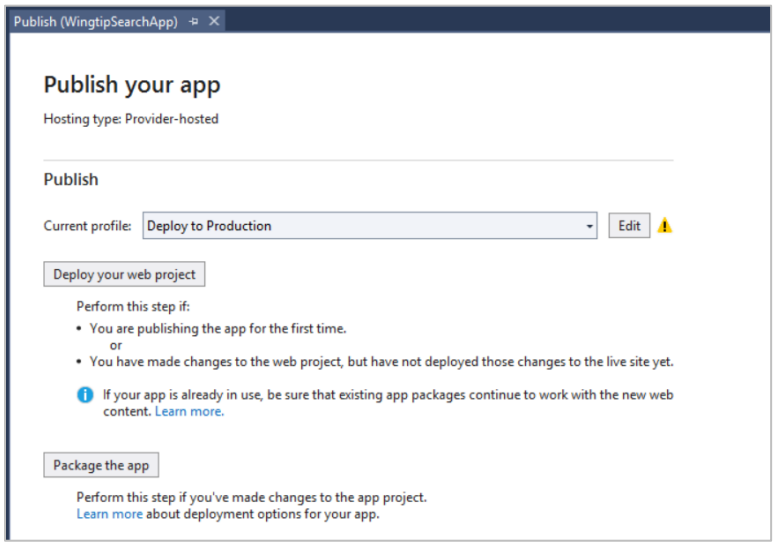
Create the App Package for the for the Wingtip Sales App

15. Create the App Package for the **WingtipSearchApp** project.

a) Right-click the the **WingtipSearchApp** project and click **Publish** to open the **Publish your app** page.



b) On the **Publish your app** page, you should see that the solution is already configured to use the **Deploy to Production** profile for the remote web deployment.

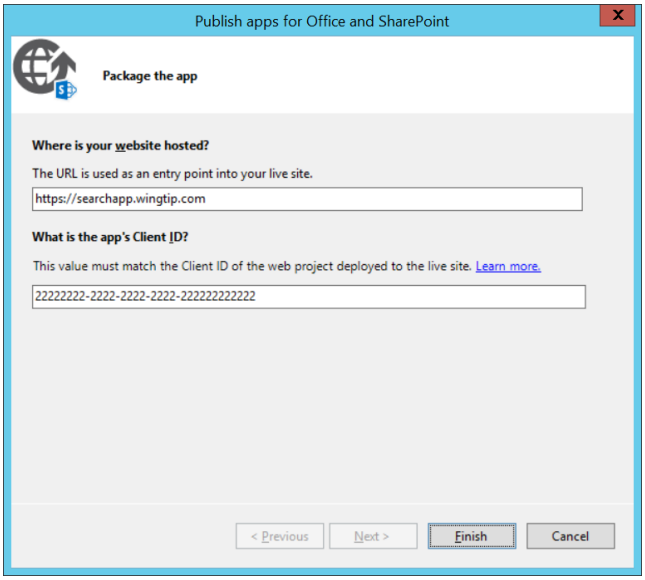


c) On the **Publish your app** page, click the **Package the app** button.

d) On the **Publish apps for Office and SharePoint** dialog, enter the following values:

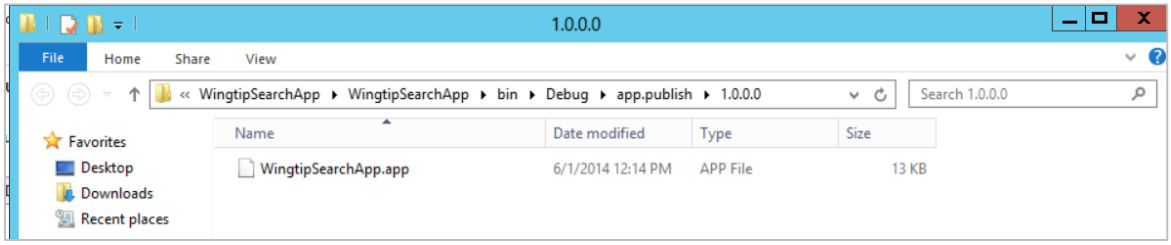
i) Enter a URL value of **https://searchapp.wingtip.com**

ii) Enter a Client ID of **22222222-2222-2222-2222-222222222222**



e) Click **Finish** to create the app package.

f) In the Windows Explorer, you should see the app package file named **WingtipSearchApp.app**. Copy the path to this file to the clipboard to make it easier to upload and publish it.

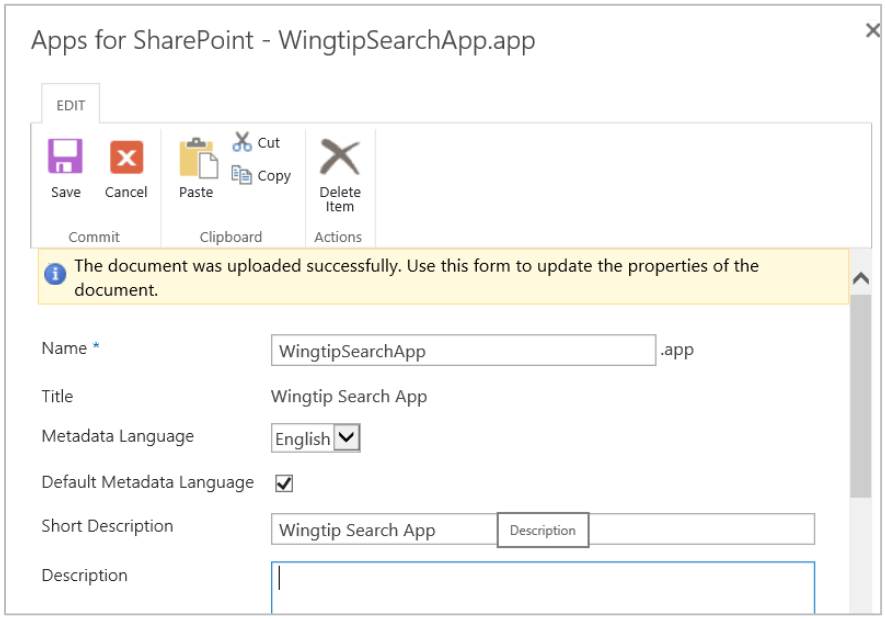


Publish the Wingtip Search App to the App Catalog

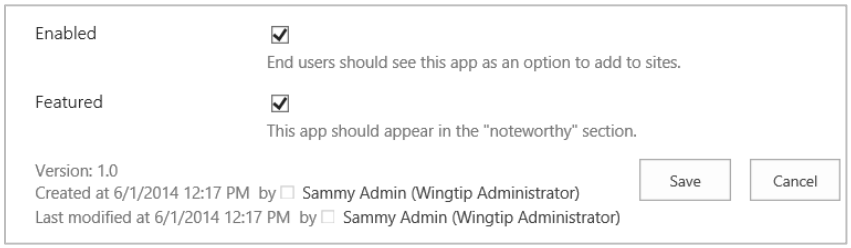
16. Publish the app to the App Catalog site collection.
    a) Navigate to the App Catalog site collection at **http://wingtipserver/sites/AppCatalog**.
    b) Click on the **Apps for SharePoint** link.
    c) Upload the app package from the following path.

```
..\Lab\StarterProjects\WingtipSearchApp\WingtipSearchApp\bin\Debug\app.publish\1.0.0.0
```

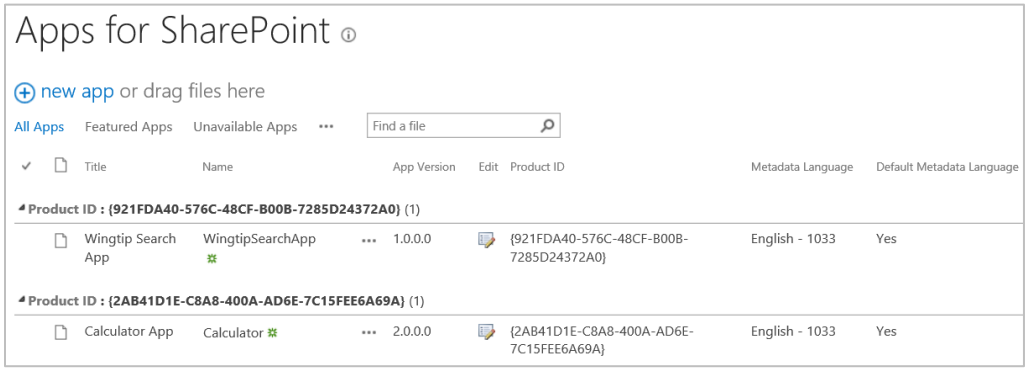
    d) In the properties page for **WingtipSearchApp.app**, add a Short Description of **Wingtip Search App**.



    e) Make the app as Featured and click **Save** to complete the app publishing process.

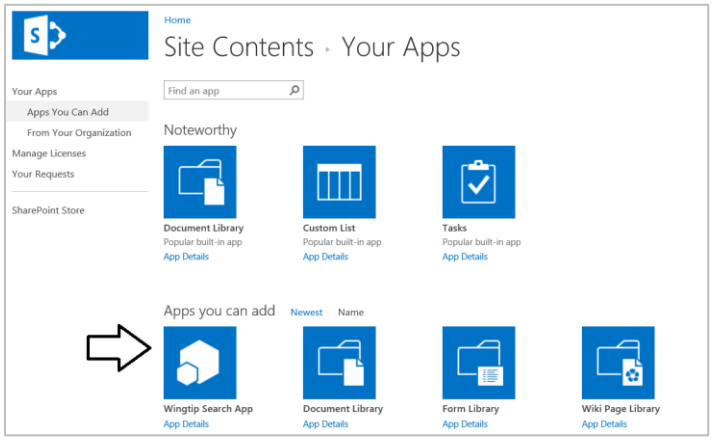


    f) You should now see the Wingtip Search App has been published along with Calculator app.

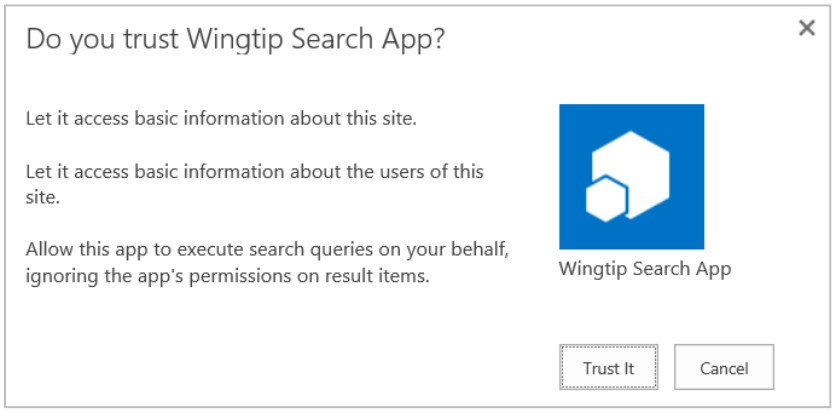


Install and Test the Wingtip Search App

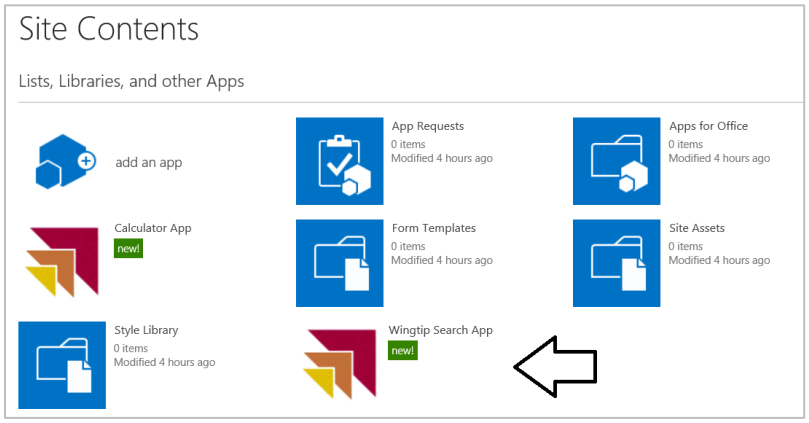
17. Install the Wingtip Sales App.
    a) Click on the **Site Contents** link in the **Quick Launch** menu to navigate to the **Site Contents** page.
    b) Click **Add an app** to navigate to the **Your Apps** page.
    c) On the **Your Apps** page, click the **Wingtip Search App** tile to begin the installation process.



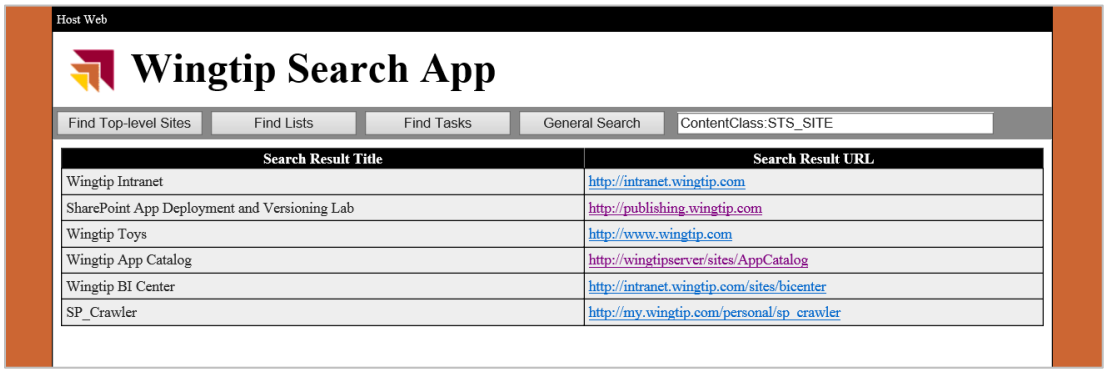
    d) When prompted to trust the app, click **Trust It**.



    e) Wait at the **Site Contents** page until you see that the app has been fully installed.



18. Test the Wingtip Sales App.
    a) Click the **Wingtip Search App** tile on the **Site Contents** page to navigate to the start page of the **Wingtip Search App**.
    b) Test the app by clicking on the **Find Top-level Sites** button.
    c) After a few seconds, you should be able to see search results in the browser.



    d) Once you can see search results, you know that the app has been deployed correctly.
    e) Click the other buttons to see what the app does.

If you are curious about the server-side C# code that is executing these searches, you can examine it in the code-behind file named **Default.aspx.cs** and in the **WingtipSearchAppWeb** project.

Student Challenge (If you have time...)

Configure the **Wingtip Search App** so that it becomes available for use from every site collection associated with the web application at with **http://wingtipserver**.