## Developing Provider-hosted Apps with Azure and O365

**Lab Time**: 60 minutes

**Lab Folder**: [c:\Student\Modules\~\Labs

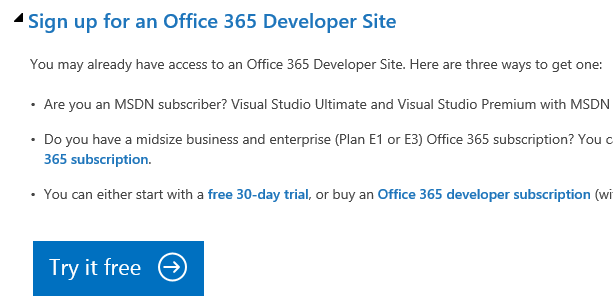
**Lab Overview**: In this lab, you will create a provider-hosted app using Azure and Office 365.

**Lab Setup**: For this lab, you must have administrator access to your own Office 365 and Azure subscriptions.

### Exercise 1: Obtain Office 365 and Azure subscriptions

In this exercise you obtain trial subscriptions to Office 365 and Azure. If you already have these subscriptions, you can skip this exercise.

1. Sign up for an Office 365 developer subscription.
   1. Navigate to <http://msdn.microsoft.com/en-us/library/office/fp179924(v=office.15).aspx>
   2. Under the heading “Sign up for an Office 365 Developer Site” click **Try It Free**.



* 1. Fill out the form to obtain your trial O365 subscription.
  2. When completed, you will have a developer site in the **[subscription].sharepoint.com domain** located at the root of your subscription (e.g. https://mysubscription.sharepoint.com)

1. Sign up for an Azure trial subscription
   1. Navigate to <https://manage.windowsazure.com>
   2. If prompted, log in using the credentials you created for your O365 subscription.
   3. After logging in, you should see a screen notifying you that you do not have a subscription

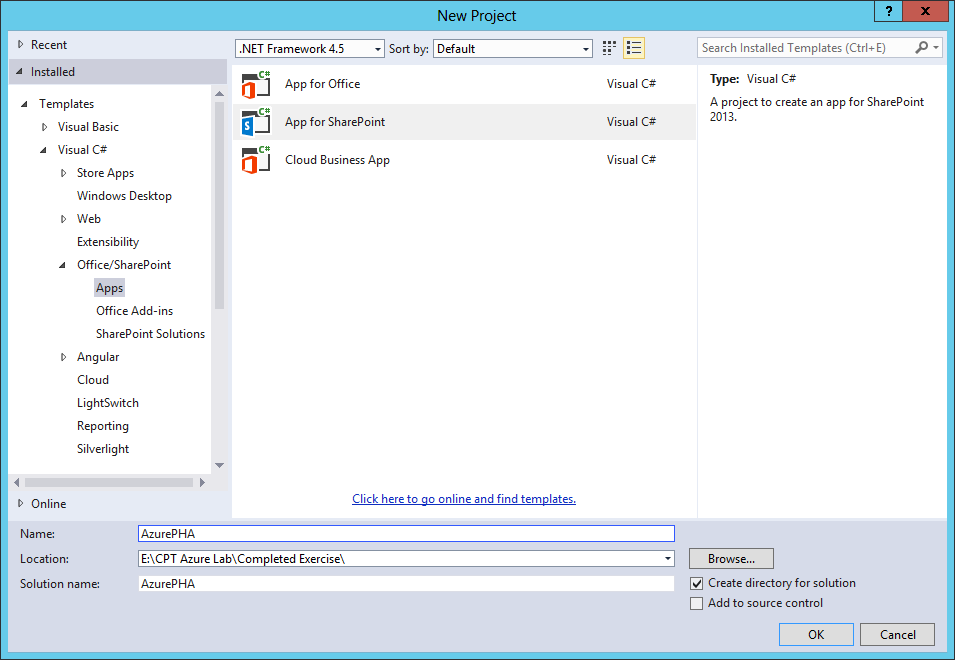


* 1. Click **Sign Up for Windows Azure**.
  2. Fill out the form to obtain your free trial.

### Exercise 2: Create a Provider-Hosted App

In this exercise you create a new provider-hosted app for your O365 subscription.

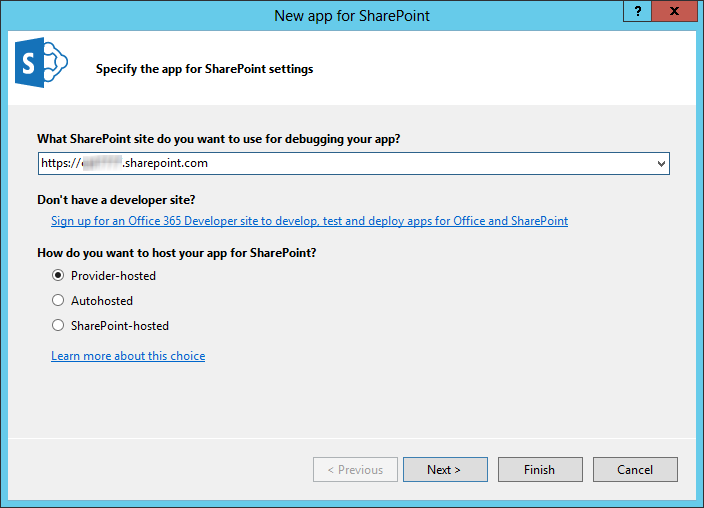
1. Create the new solution in Visual Studio 2013:
   1. Launch **Visual Studio 2013** as administrator:
      1. **Windows** Keyboard Key 🡪 Right click on the Visual Studio 2013 tile and select **Run as administrator**.
   2. In Visual Studio select **File 🡪 New 🡪 Project**.
   3. In the **New Project** dialog:
      1. Select **Templates 🡪 Visual C# 🡪Office/SharePoint🡪 Apps**.
      2. Click **App for SharePoint 2013**.
      3. Name the new project **AzurePHA** and click **OK**.



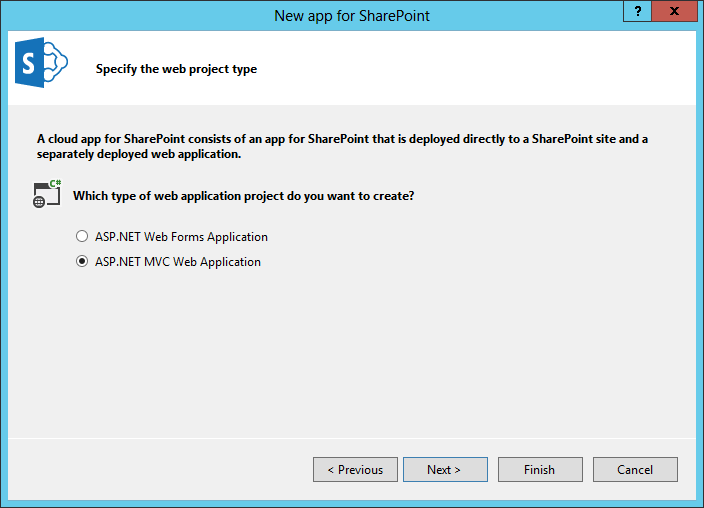
* 1. In the New App for SharePoint wizard:
     1. Enter the address of a SharePoint site to use for testing the app

NOTE: The targeted site must be based on a Developer Site template

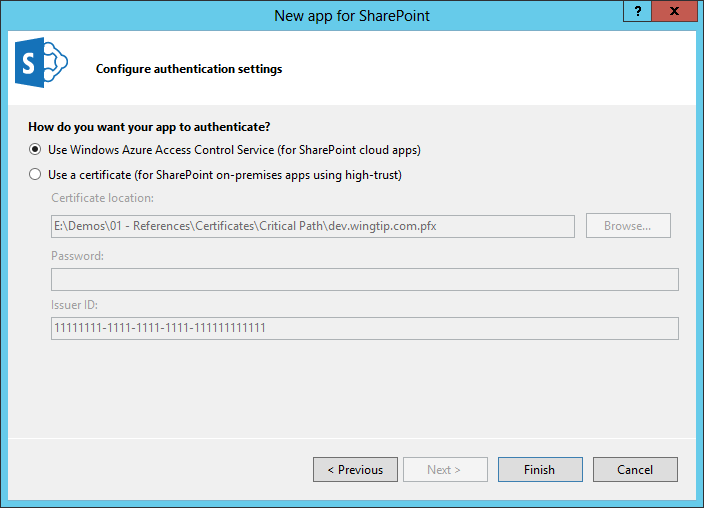
* + 1. Select **Provider-Hosted** as the hosting model.
    2. Click **Next**.



* + 1. Select **ASP.NET MVC Web Application**.
    2. Click **Next**.

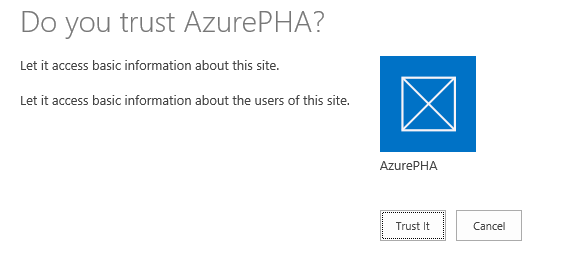


* + 1. Select the option labeled **Use Windows Azure Access Control Service (for SharePoint cloud apps)**.
    2. Click **Finish**.

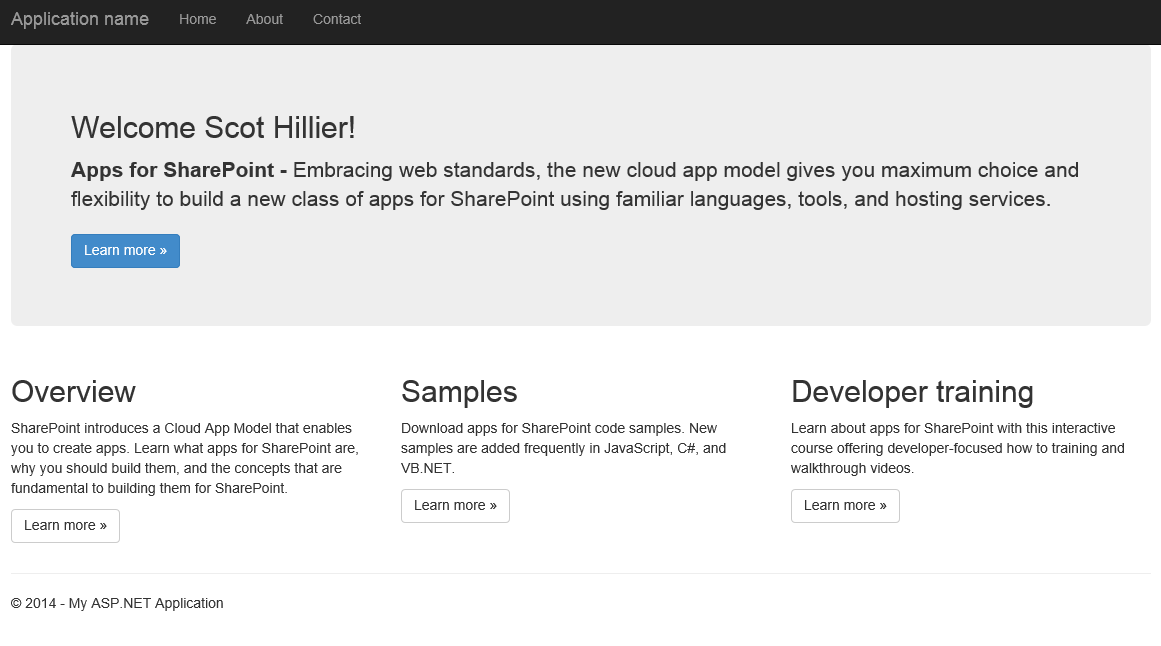


* + 1. When prompted, log in using your O365 administrator credentials.

1. Test your app
   1. Press **F5** to begin debugging.
   2. When prompted, log in using your O365 administrator credentials.
   3. When prompted, click **Trust it**.



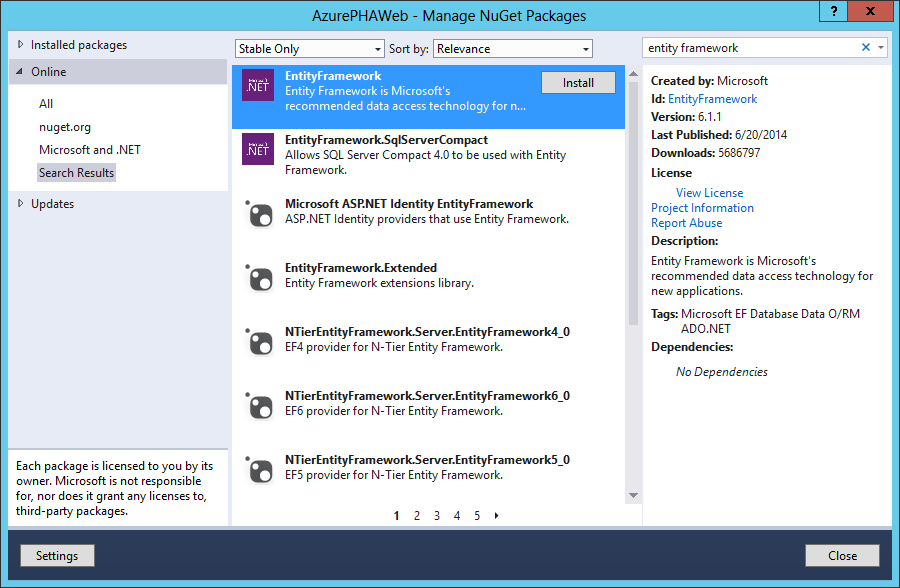
* 1. Verify that the app home page shows and that it properly welcomes you by name.



### Exercise 3: Access a Database using MVC5

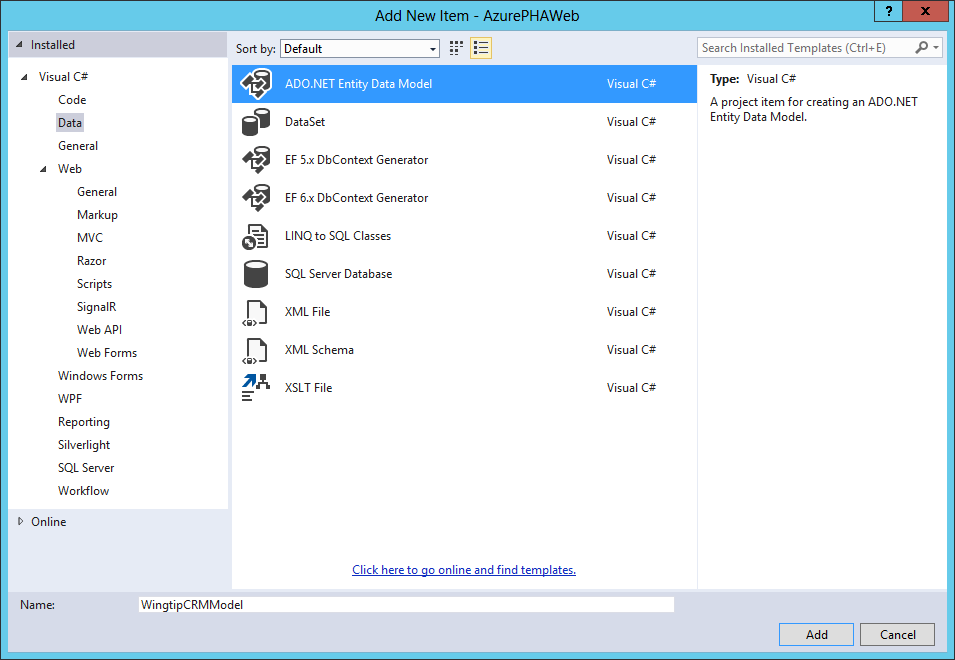
In this exercise, you will add additional functionality to the app to read data from the Wingtip CRM database, which was set up in previous labs.

1. Update the Entity Framework package.
   1. Right click the **AzurePHAWeb** project and select **Manage NuGet Packages**.
   2. Type **Entity Framework** in the search box.
   3. Click the **Install** button for Entity Framework version 6.

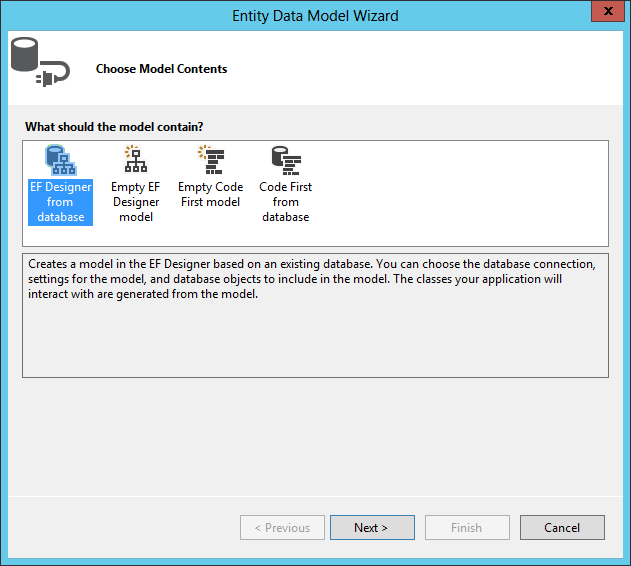


* 1. After the package is installed, click **Close**.

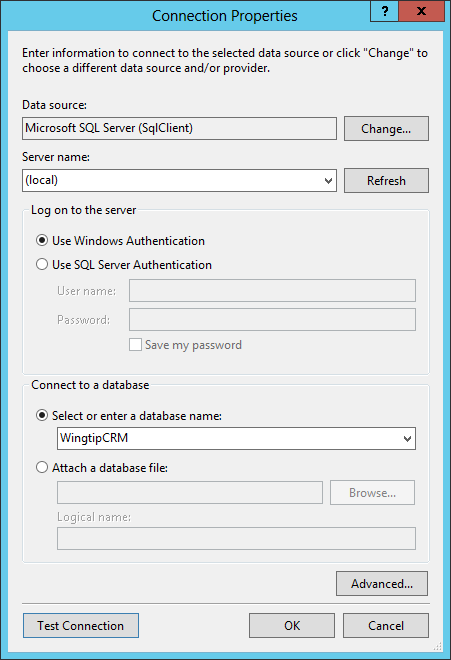
1. Add an Entity Framework model.
   1. In the **Solution Explorer**, right-click the **Models** folder in the **AzurePHAWeb** project.
   2. Select **Add 🡪 New Item** from the context menu.
   3. In the New Item dialog:
      1. Select **Visual C# 🡪 Data 🡪 ASP.NET Entity Data Model**.
      2. Name the new model **WingtipCRMModel.edmx**.
      3. Click **Add**.



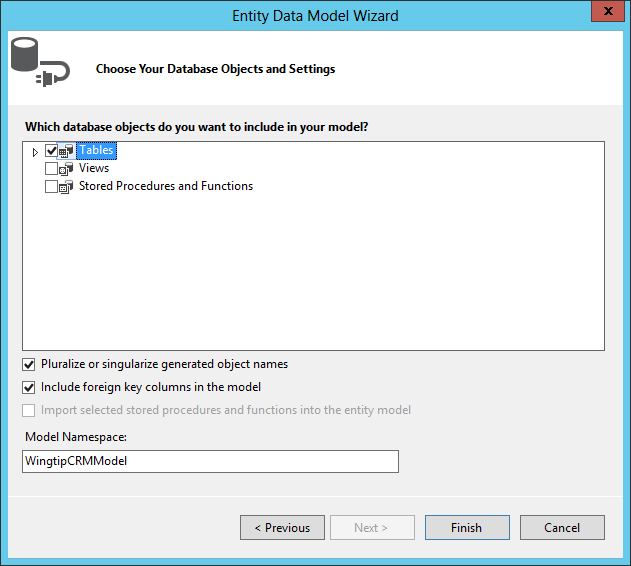
* 1. In the Entity Data Model wizard:
     1. Click **EF Designer from Database**.
     2. Click **Next**.



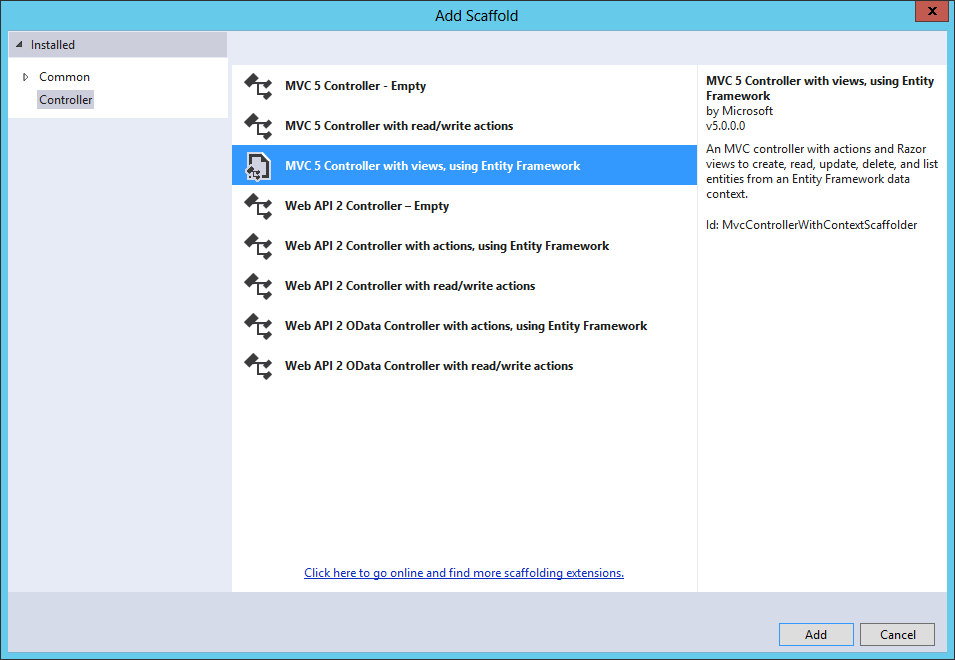
* + 1. Click **New Connection**.
    2. In the Connection Properties dialog:
       1. Enter **(local)** in the **Server Name** field.
       2. Enter **WingtipCRM** in the **Database Name** field.
       3. Click **Test Connection**.
       4. Click **OK**.
    3. Click **Next**.



* + 1. Check **Tables**.
    2. Click **Finish**.

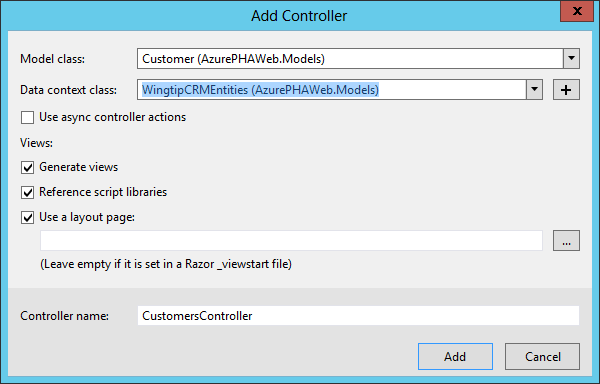


1. Add a controller.
   1. **Build** the AzurePHAWeb project.
   2. Right-click the Controllers folder and select **Add 🡪 Controller**.
      1. Select **MVC5 Controller with views using Entity Framework**.
      2. Click **Add**.



* + 1. Select **Customer** as the Model Class.**.**

Select **WingtipCRMEntities** as the Data Context Class.

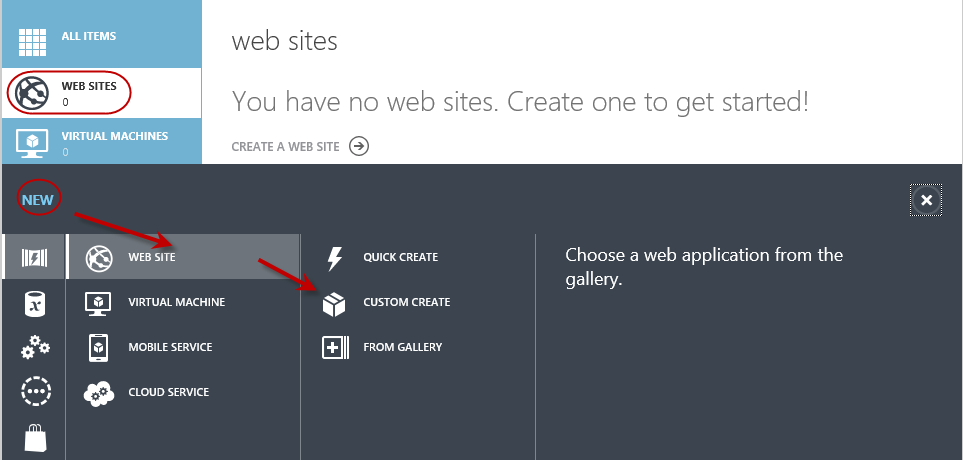
* + 1. Click **Add**.
    2. 

1. Update the App Manifest
   1. In the **AzurePHA** project, double-click the **AppManifest.xml** file.
   2. Update the Start Page to be **AzurePHAWeb/Customers**.
2. Test your app
   1. Press **F5** to begin debugging.
   2. When prompted, log in using your O365 administrator credentials.
   3. When prompted, click **Trust it**.
   4. Verify that the customer data appears in the app.

### Exercise 4: Deploy the App to Production

In this exercise, you will deploy the database and app to the O365/Azure environment.

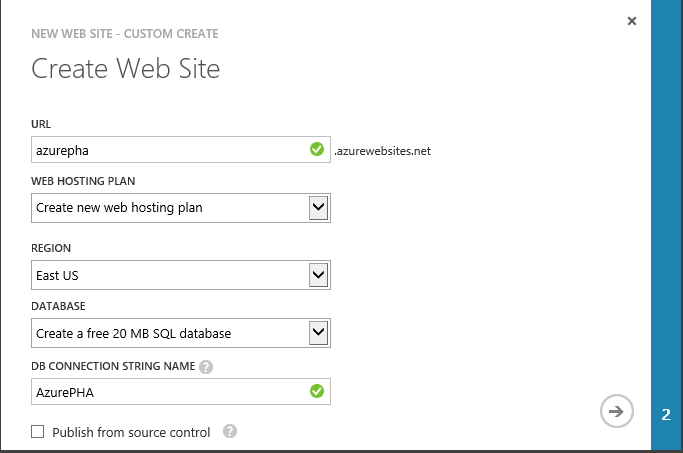
1. Create a Web Site and SQL Azure database
   1. Log into <https://manage.windowsazure.com> as an administrator.
   2. Click **Web Sites**.
   3. Click **New**.
   4. Click Custom Create.



* 1. Enter a URL for the application.

NOTE: URLs must be globally unique, so you will have to choose one not used by another.

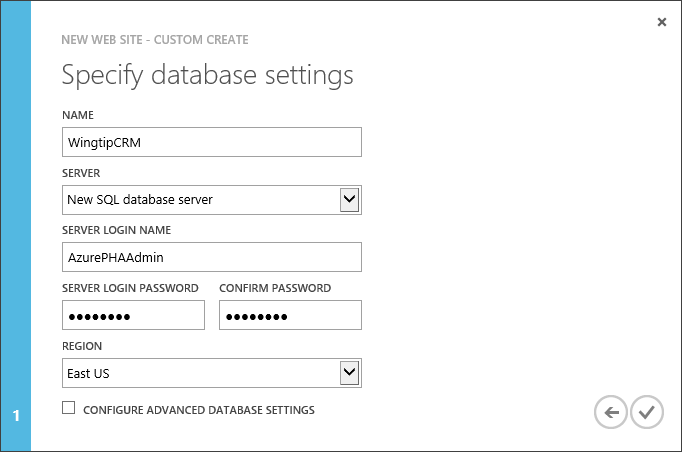
* 1. Select **Create New Web Hosting Plan**.
  2. Select an appropriate **Region**.
  3. Select **Create a free 20MB SQL Database**.
  4. Name the database connection string **AzurePHA**.
  5. Click the **Right Arrow**.



* 1. In the Specify Database Settings
     1. Name the new database **WingtipCRM**.
     2. Select **New SQL database server**.
     3. Name the administrator AzurePHAAdmin and enter a password.

Write down the credentials for later!

* + 1. Pick an appropriate **Region**.
    2. Click the **checkmark**.



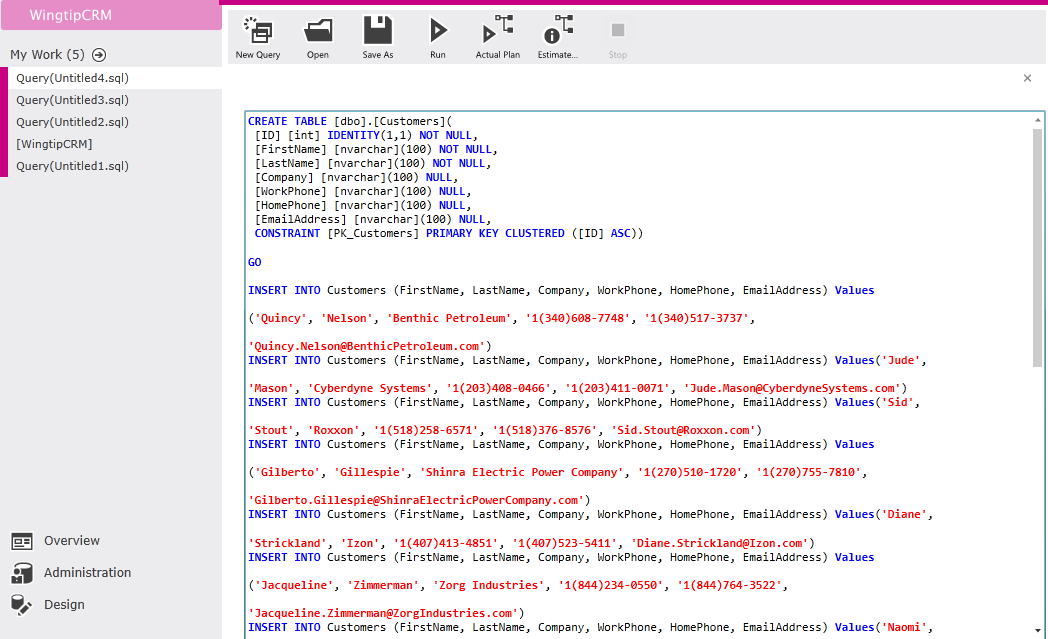
1. Upload test data to SQL Azure:
   1. In the Azure portal, click **SQL database**.
   2. Click **WingtipCRM**.
   3. Click **Run Transact SQl Queries Against Your Database**.
   4. When prompted to add a firewall rule, click **Yes**.



* 1. When prompted, select to manage the WingtipCRM database.



* 1. Log in to the database server using the credentials you created earlier.
  2. Paste the contents of the script **CreateAzureWingtipCrmDatabase.sql** into the query window.
  3. Click **Run**.



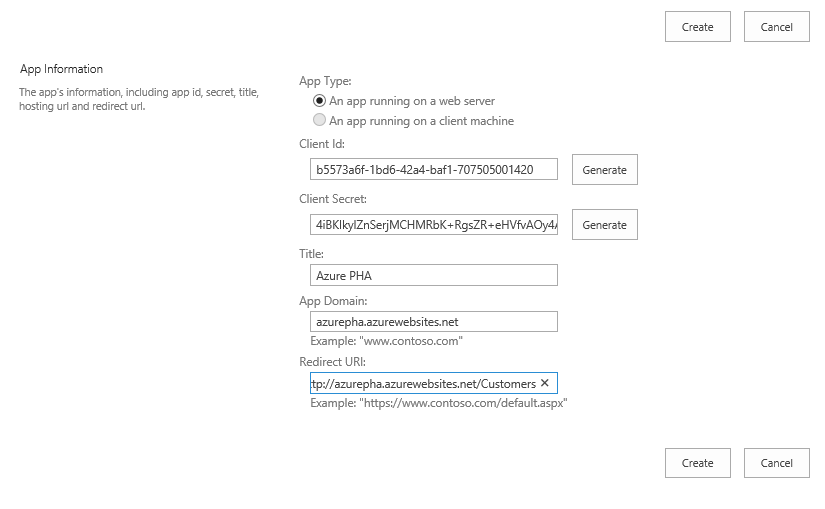
1. Update the SQL Azure connection string in the provider-hosted app.
   1. In the Azure portal, click **SQL database**.
   2. Click **WingtipCRM**.
   3. Click **View SQL database connection strings**.
   4. Copy the **ADO.NET** connection string.
   5. Open the **web.config** file for the **AzurePHAWeb** project.
   6. Locate the part of the connection string surrounded by encoded quotes as shown:



* 1. Carefully replace the information between the encoded quotes with the connection string you obtained from the Azure portal.

Be sure to update the string with your password as appropriate.

1. Register the app in Office 365
   1. Log into the O365 developer site as an administrator
   2. From the developer site, navigate to **/\_layouts/15/appregnew.aspx**.
   3. Click **Generate** next to **Client ID**.
   4. Click **Generate** next to **Client Secret.**
   5. Enter **Azure PHA** as the **Title**.
   6. Enter the **App Domain** for the Azure web site you created earlier (e.g., azurepha.azurewebsites.net)
   7. Enter the Redirect URI as the reference for the Customers page (e.g. <https://azurepha.azurewebsites.net/Customers>).
   8. Click **Create**.



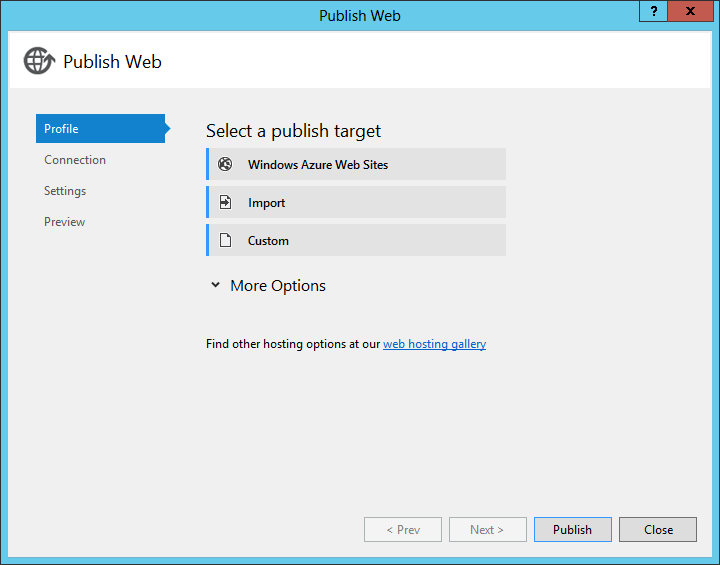
* 1. **Save** the Client ID and Client Secret separately for later use.

1. Update the provider-hosted app
   1. In the **AzurePHA** project open the **AppManifest.xml** file in a text editor.
   2. Update the Client ID and App Start page to reflect the values you created earlier.



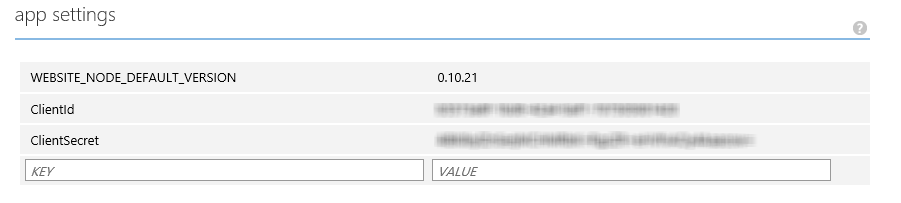
* 1. Open the **web.config** file for the **AzurePHAWeb** project.
  2. **Update** the Client ID and Client Secret to use the generated values.

1. Publish the remote web
   1. Right click the **AzurePHAWeb** project and select **Publish**.
   2. Click **Windows Azure Web Sites**.



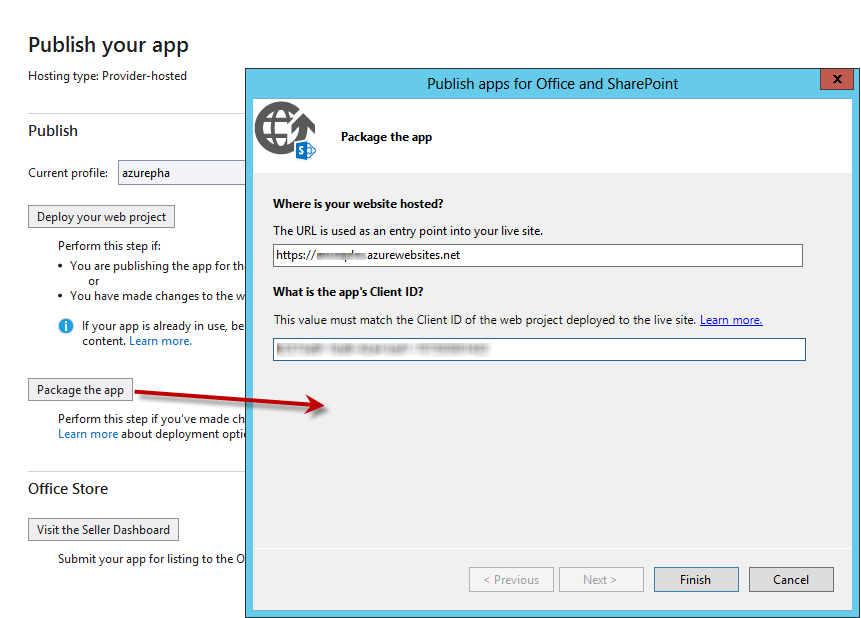
* 1. When prompted, select to deploy the remote web to the existing Azure web site you created earlier.
  2. **Publish** the remote web.

1. Update information in the Azure Portal
   1. Return to the **Azure Management portal**.
   2. Click **Web Sites**.
   3. **Select** your Azure Web Site.
   4. Click **Configure**.
   5. In the **App Settings** section, add a **ClientId** and **ClientSecret** setting.
   6. Set the values to the values you generated earlier.
   7. Click **Save**.

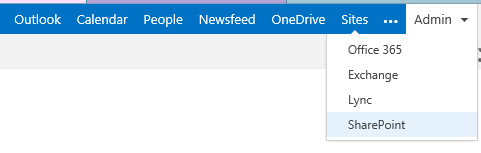


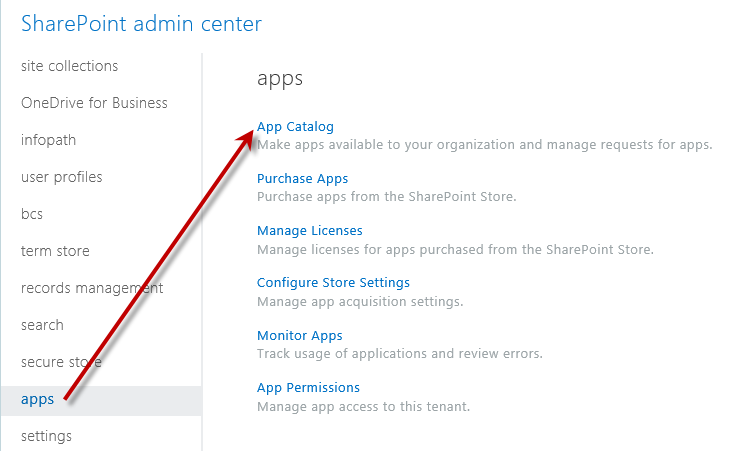
The update to the Azure Portal is required because the Client ID and Client Secret are not automatically picked up from the web.config file in Azure.

1. Package the SharePoint App
   1. Right click the **AzurePHA** project and select **Publish**.
   2. Click **Package the App**.
   3. **Enter** the Start URL and Client ID for the app.
   4. Click **Finish**.



1. Publish the App to the Corporate Catalog
   1. Return to the O365 tenant and select **Admin🡪SharePoint**.

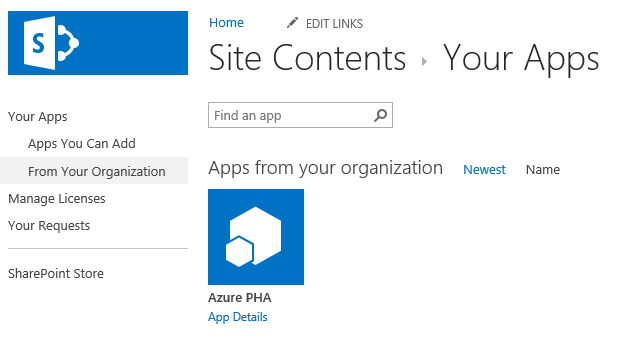


* 1. Click **Apps 🡪 App Catalog**.
  2. 
  3. Select **Create new app catalog site**.
  4. Click **OK**.
  5. Fill out the required information for the new app catalog site and click **OK**.
  6. Once created, navigate to the new app catalog site.
  7. In the app catalog site, click **Apps for SharePoint**.
  8. Click **New**.
  9. **Browse** to the app package you created earlier.
  10. Add the app package to the Apps for SharePoint library.

1. Add the app to a SharePoint site
   1. Navigate to a site in your O365 tenancy.
   2. Click **Site Contents**.

NOTE: If you are using the Developer site, it may have an older version of the app still installed from testing. You must remove the app from the site AND remove the entry from the “Apps in Testing” list or the new app will not install..

* 1. Click **Add an App**.
  2. Click **From Your Organization**.



* 1. Click the app installer.
  2. When prompted, click **Trust It**.

1. Test the App
   1. Use the tile to **launch** the app.
   2. Verify that data from the SQL Azure database appears in the app.
   3. Manually remove **/Customers** from the URL in the browser. This should navigate you to the home page where you should be welcomed validating that the app communicates with SharePoint.

Congratulations! You have completed building a provider-hosted app using Office 365 and Azure.