Publishing

This tutorial shows you the steps on how to publish the Excel Colorizer Add-in for Excel 2016 from the west U.S. for distribution in Office Store. Visual Studio Community 2015 is the IDE of choice. Microsoft Azure is the web hosting platform for the Colorizer.

To publish your add-in using other methods that are specific to your scenario, please see MSDN resources on publishing.

Topics

Decide on the Office Add-ins distribution end-points

Decide on the distribution end-points to publish your Office Add-ins:

a. Office Store
b. Office Add-ins catalog on SharePoint
c. Exchange catalog
d. Network shared folder add-in catalog

This tutorial uses Microsoft Azure to host the add-in and submit the add-ins to the Office Store for publishing.

Set up your development computer with Azure SDK for .NET, an Azure subscription, and Office 2016

1. Install the Azure SDK for .NET from the Azure downloads page. This tutorial uses the free Microsoft Visual Studio Community 2015.
   a. Under **Languages**, choose **.NET**.
   b. Choose the version of the Azure .NET SDK that matches your version of Visual Studio, if you already have Visual Studio installed.
   c. When you're asked whether to run or save the installation executable, choose **Run**.
   d. In the Web Platform Installer window, choose **Install**.

2. Install Office 2016. (The Colorizer add-in works only in Excel 2016 or Excel Online.)

Note
You can get a trial version of Office for one month.

3. Get your Azure account.

Note
If you're a Microsoft Developer Network (MSDN) subscriber, you get an Azure subscription as part of your MSDN subscription.

If you're not an MSDN subscriber, you can still get a free trial of Azure at the Windows Azure website.

Create a website in Azure

There are a couple of ways you can create an empty Azure website. If you're using Visual Studio 2015 or the free community version, follow the steps below to create an Azure website from within the Visual Studio IDE.

Using Visual Studio Community 2015

1. In Visual Studio, in the **View** menu choose **Server Explorer**. Right click **Azure** and choose **Connect to Microsoft Azure subscription**. Follow the instructions for connecting to your Azure subscription.

2. In Visual Studio, in **Server Explorer**, expand **Azure**, right-click **App Service**, and then choose **Create New App Service**.

3. In the **Create Web App on Windows Azure** dialog box, provide this information:
   - Enter a unique **Web App name** for your site. Azure verifies that the site name is unique across the azurewebsites.net domain.
   - Choose the **App Service plan** you're using to authorize creating this website. If you create a new plan, you also need to name it.
   - Choose the **Resource group** for your site. If you create a new group, you also need to name it.
   - Choose a geographical **Region** appropriate for you.
   - For **Database server**: accept the default of No database and then choose **Create**.
     The new website appears under the chosen resource group under **App Service** under **Azure** in **Server Explorer**.

4. Right-click the new website, and then choose **View in Browser**. Your browser opens and displays a webpage with the message "This web site has been successfully created."

5. In the browser address bar, change the URL for the website so that it uses HTTPS and press **Enter** to confirm that the HTTPS protocol is enabled. The Office Add-in model requires add-ins to use the HTTPS protocol.

6. In Visual Studio 2015, right-click the new website in **Server Explorer**, choose **Download Publish Profile** and then save the profile to your computer. The publish profile contains your credentials and enables you to publish your Office Add-in to the Azure website.

Publish your Office Add-in to the Azure website

1. With your add-in open in Visual Studio, expand the solution node in **Solution Explorer** so that you see both projects for the solution.

2. Right-click the web project, and then choose **Publish**.

   The web project contains Office Add-in website files so this is the project that you publish to Azure.

3. In **Publish Web**, choose **Import**.

4. In **Import Publish Settings**, choose **Browse**, and then browse to the place where you saved your publish profile earlier in this topic. Choose **OK** to import your profile.

5. In **Publish Web**, on the **Connection** tab, accept the defaults and choose **Next**.

   Choose **Next** > again to accept the default settings.

6. On the **Preview** tab, choose **Start Preview**. The preview shows you all the files in the web project that will be published to the Azure website.

7. Choose **Publish**. Visual Studio publishes the web project for your Office Add-in to your Azure Web Site.

8. When Visual Studio finishes publishing the web project, your browser opens and shows a webpage with the text "This web app has been successfully created." This is the current default page for the website.

   To see the webpage for your add-in, change the URL to use https: and add the path of your add-in's default HTML page. For example, the changed URL should look like https://YourDomain.azurewebsites.net/Addin/Home/Home.html. This confirms that your add-in's website is now hosted on Azure. Copy this URL because you'll need it when you edit the add-in manifest file later in this topic.

Edit the add-in manifest file to point to the Office Add-in on Azure

1. In Visual Studio with the Office Add-in open in **Solution Explorer**, expand the solution so that both projects are shown.

2. Open the add-in manifest XML file.

3. For **Source Location**: enter the URL for the add-in's main HTML page that you copied in the previous step after you published the add-in, for example,
   https://YourDomain.azurewebsites.net/Addin/Home/Home.html.
   Save the manifest file.

Run the add-in in the Office client application

By now, your source location has changed from the local server for the testing environment to the website URL in Azure.

Run your add-in in the Office client and test that the add-in is working.

Submit your add-in to the Office Store

The recommended reading for submitting your add-in to the Office Store is as follows:

Submit Office and SharePoint Add-ins and Office 365 web apps to the Office Store

Checklist for submitting Office and SharePoint Add-ins and Office 365 web apps to the Seller Dashboard

Office Store app and add-in submission FAQ

Validation policies for apps and add-ins submitted to the Office Store (version 1.9)

Office Add-ins XML manifest

Update the version of your JavaScript API for Office and manifest schema files

Referencing the JavaScript API for Office library from its content delivery network (CDN)

To publish the Excel Colorizer Add-in for Excel 2016, the following tasks are performed:

1. To include your app or add-in in the Office Store, submit it to the Microsoft Seller Dashboard.

   Create an individual or company account and, if applicable, add payout information. To open an account, see Create a seller account and add payout information.

   For this tutorial, an individual account is created to publish the add-in. Because the add-in is free, no payout information is provided.

   Fill out the account profile, submit, and wait for the approval.

2. An icon, App logo, screenshots, and .html files containing privacy and support statements are prepared before the submission. Follow the specifications for the image dimensions in the Checklist for submitting Office and SharePoint Add-ins and Office 365 web apps to the Seller Dashboard.

3. Publish the icon, privacy and support statements to your Azure account.

4. In the manifest file, add the IconUrl, SupportUrl, and the Requirements elements in addition to the default elements. (The Excel Colorizer uses the new Excel add-in JavaScript APIs that runs in Excel 2016 or Excel Online). The manifest schema version is 1.1.

Note
*Every add-in is different, to ensure that your add-in works as expected, see Specify Office hosts and API requirements and Office add-in requirement sets .

5. Reference the Microsoft hosted Office.js file from its CDN URL. Don't include a copy of the Office.js file in your add-in. See Office Store app and add-in submission FAQ > How do I reference the JavaScript APIs for Office in my add-ins?

6. If you're using Visual Studio, validate the manifest by clicking on **Build** > **Publish** > **Perform Validation Check**

7. Login to the Microsoft Seller Dashboard to add the App.
   - Enter the version number. The version number submitted via the Seller Dashboard must be the same as that in the add-in manifest. For example:

     Seller Dashboard: 1.0.0.0
     Add-in Manifest: 1.0.0.0

   - Upload the screenshots and the App logo.

   - Upload the app package (i.e. the manifest XML file).

   - Supply the Support and Privacy Statement URL. Include your contact information in the Support Statement URL or alternatively within your add-in. This can be any or all of the following: email address, phone number, contact form, valid social media link (e.g. a URL to your contact page on Facebook or Twitter), or comments form.

8. Submit the app and wait for the approval.

9. After your add-in is submitted, a Validation Test Results report will be returned to you if the add-in is not approved by the Office Store. The validation test is performed based on Validation policies for apps and add-ins submitted to the Office Store (version 1.9), and lists the required changes. Make the changes and resubmit your add-in.

Resources

Publishing basics

Host an Office Add-in on Microsoft Azure

Submit Office and SharePoint Add-ins and Office 365 web apps to the Office Store