Publish ASP.NET MVC Core Application to Azure Web App

Introduction

This lab is part of a series. This third lab will show you how to publish a simple ASP.NET Core application to the Azure Web App that you created and configured in the previous labs.

Terminology

The Portal uses a user interface concept that tends to expand horizontally towards the right. Every time that you choose something, rather than popping open a dialog box, it creates a new panel of in the user interface. These panels are called **blades**. I'll be referring to UI blades through this lab.

Variables

A lot of the resources that you create in this lab are going to need unique names. When I say unique, I mean that they're going to need to be unique for Azure and not just fun and creative. Since I can't possibly know which values that you're going to need to choose, I'm going to give you the list of these values now and let you choose them. I'll refer to these as "variables" throughout the lab and when I refer to them, I'll put them in squiggle brackets like this – {{Variable Name}}.

Variable Name	Description	Your Value
{{App Name}}	This is the name of your application in Azure. This will eventually turn into the URL for your application. For example, if my App Name is 'thingy123' application URL that azure generates will be https://thingy123.azurewebsites.net.	
{{Resource Group}}	This is the name of the Azure resource group.	
{{App Service URL}}	This is the URL for your web app. This value is generated for you by Azure.	https://{{App Name}}.azurewebsites.net
{{Client Id}}	This is the MSA application Id that's generated for you	
{{Client Secret}}	This is the MSA application secret that's generated for you	

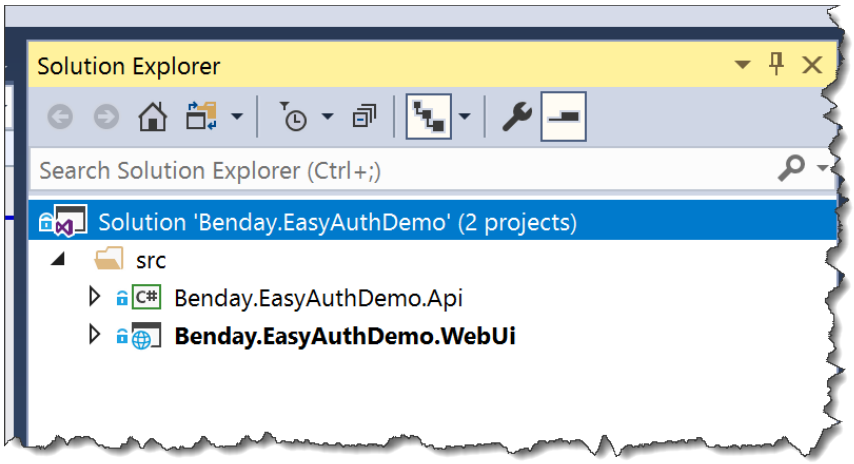
Source Code

You can download the source code for this lab from
https://www.benday.com/labs/azure-web-app-security-2018/benday-azure-web-app-code-lab3.zip

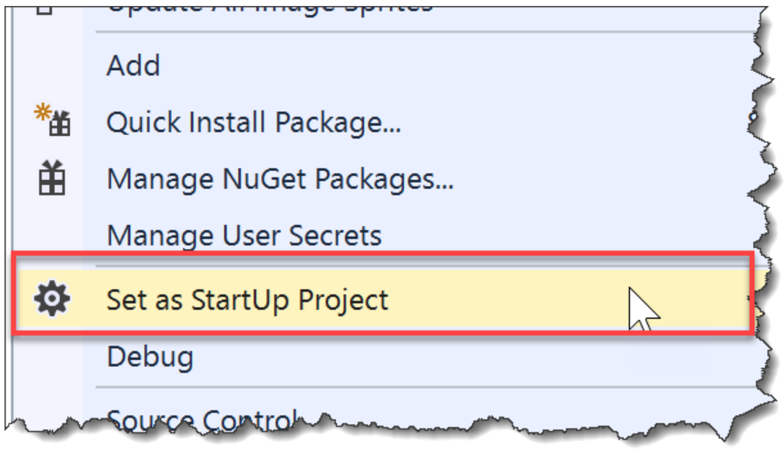
Open the Sample Solution & Run It Locally

For this lab, you're going to use a simple ASP.NET MVC Core application that's in the zip file for lab 3. This code is very simple. It's not much more than what you'd get if you created a new solution and ASP.NET MVC Core project.

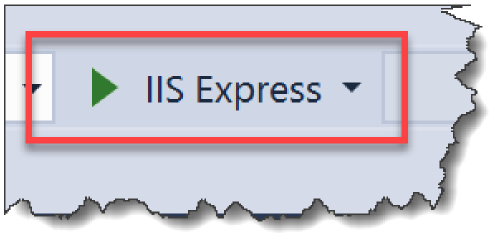
1. Locate the zip file for this lab.
2. Extract the zip to a folder on your local disk (for example, c:\temp\azure-labs)
3. In the **before** folder for this lab, open the **Benday.EasyAuthDemo.sln** solution using Visual Studio 2017. When it's opened, you should see two projects in Solution Explorer.


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Benday.EasyAuthDemo' (2 projects)
src
Benday.EasyAuthDemo.Api
Benday.EasyAuthDemo.WebUi

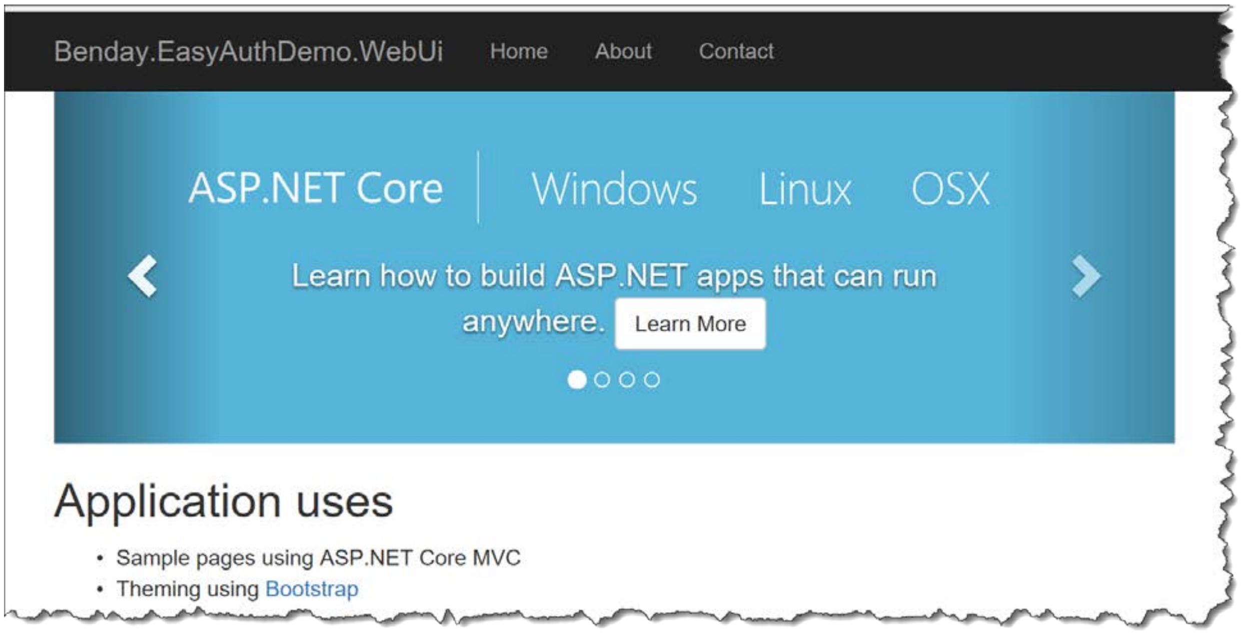
4. Let's make sure that the web project is marked as the start up project. In **Solution Explorer**, **right-click** on the **Benday.EasyAuthDemo.WebUi** project. From the context menu, choose **Set as StartUp Project**.



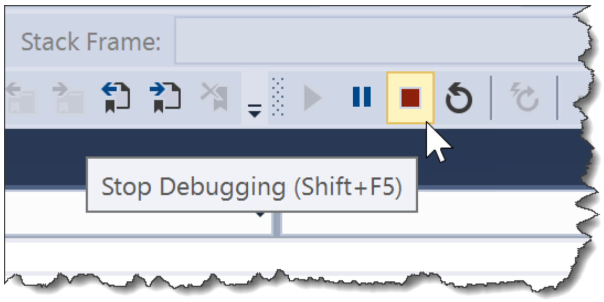
5. Start the application locally by clicking on the **IIS Express** button in the main menu of Visual Studio.



6. The application should start in a browser and should look something like the image below.


Benday.EasyAuthDemo.WebUi
Home
About
Contact
ASP.NET Core
Windows
Linux
OSX
Learn how to build ASP.NET apps that can run anywhere.
Learn More
Application uses
Sample pages using ASP.NET Core MVC
Theming using Bootstrap

7. Click around in the application a little bit just to familiarize yourself with it. There's not a whole lot in this application. But definitely notice that you didn't have to enter any username or password to run the application.
8. Go back to the Visual Studio window. Click the **Stop Debugging** button in the toolbar menu.



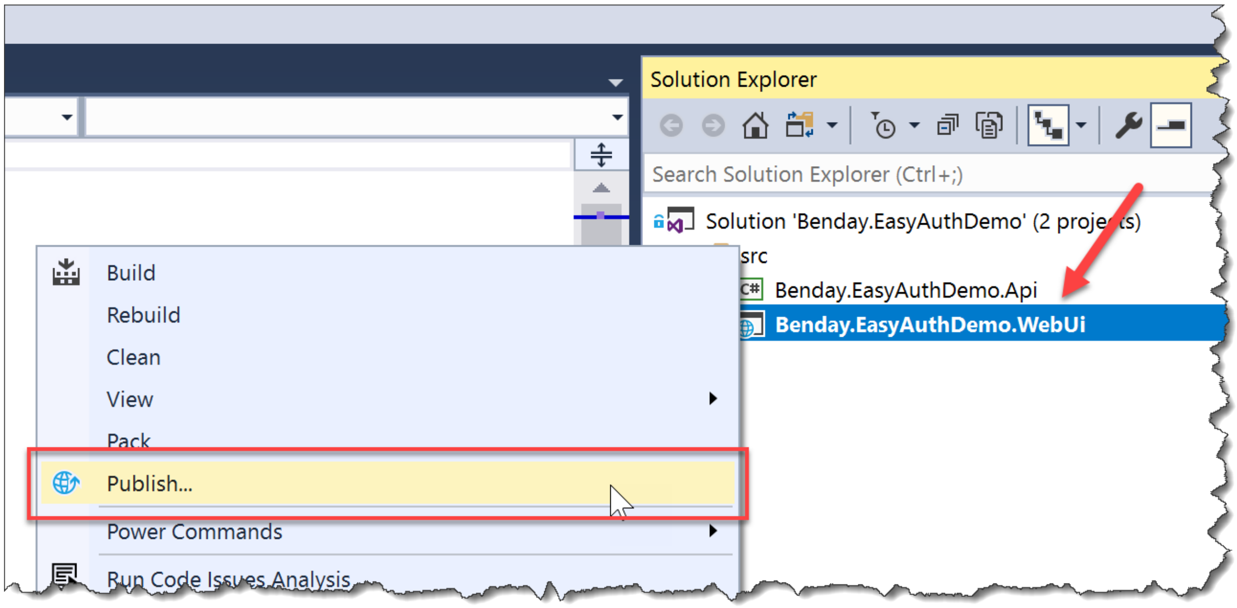
Deploy Your Code to Azure

Now that you've played with the application a little bit locally, let's publish the app up to your Azure Web App.

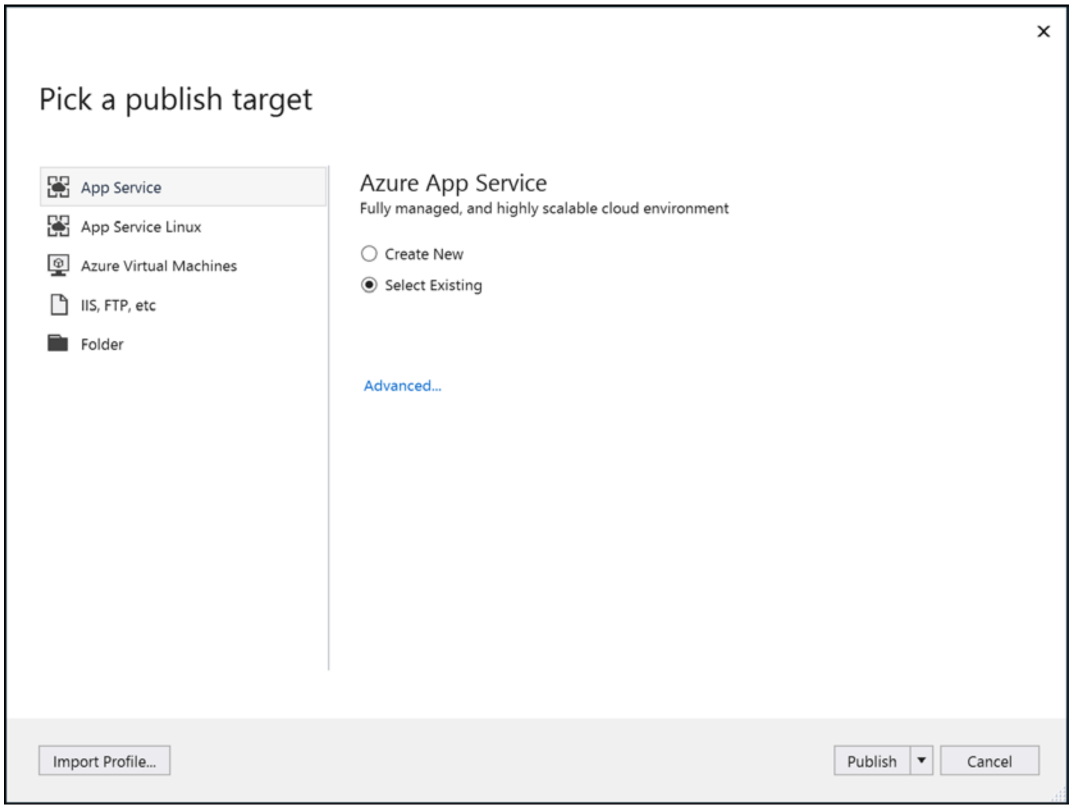
Before we do the deploy, I want to point out that, yes, I know the phrase "friends don't let friends 'right-click → deploy'". For production applications and for DevOps and happiness and everything that good in the world, you shouldn't do a Right-Click → Deploy from Visual Studio. This is just a demo app. In the real world, please don't do this. Please deploy from an automated build and/or an automated release. If you want to learn more about that, check out my Pluralsight courses on VSTS and DevOps.

Ok. Now let's do something that's dirty and evil. Let's do a Right-click → Deploy from Visual Studio.

9. In **Solution Explorer**, right-click on the **Benday.EasyAuthDemo.WebUi** project. From the context menu, choose **Publish…**.



10. You should now see a dialog that asks you to **Pick a publish target.** Ensure that **App Service** is selected. Under **Azure App Service**, choose **Select Existing**. Click the **Publish** button.

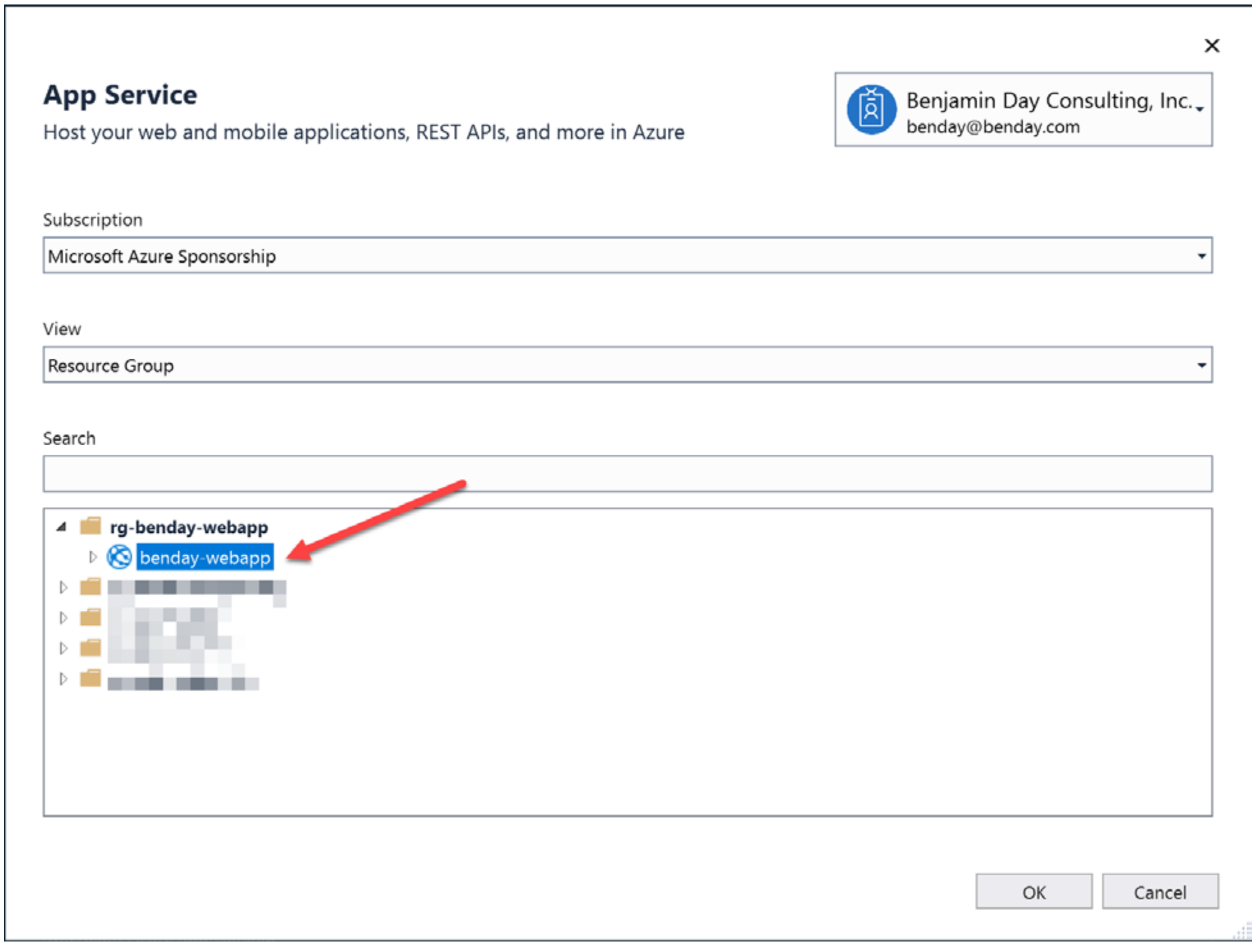


11. If prompted, log in using the same credentials you use to access the Azure Portal.

12. You should see a dialog named **App Service** that walks your through the process of choosing the Web App that you want to deploy to. Choose your **Subscription**. Set the **View** to **Resource Group**.

    You should see the list of resource groups in your subscription. Locate the Azure App Service Web App that you created before. (HINT: it should be called **{{App Name}}**).

    Select your **web app** and click the **OK** button.



App Service

Host your web and mobile applications, REST APIs, and more in Azure

Benjamin Day Consulting, Inc.
benday@benday.com

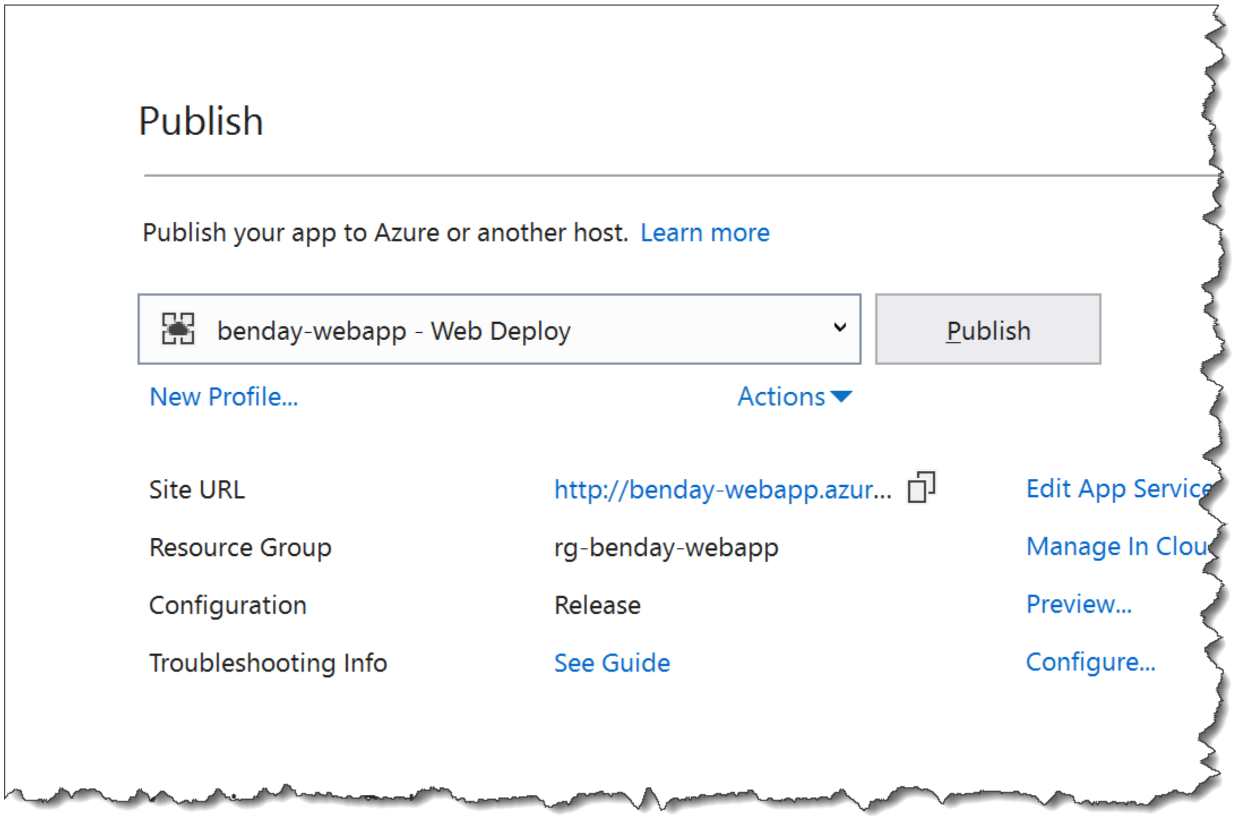
Subscription: Microsoft Azure Sponsorship

View: Resource Group

Search

rg-benday-webapp
benday-webapp

13. You'll see a dialog in Visual Studio that looks like the image below. Meanwhile, Visual Studio will be doing some thinking while it deploys your application to Azure.



Publish

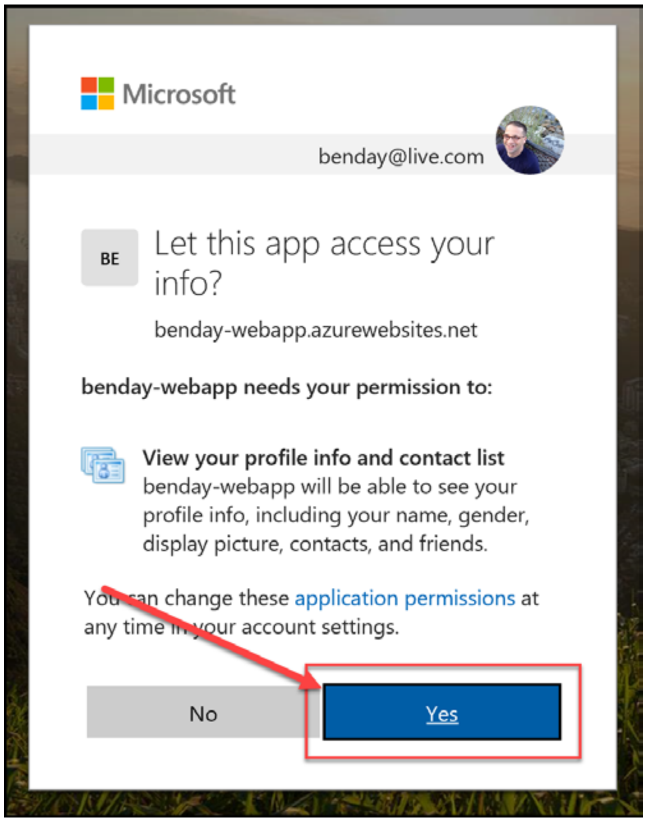
Publish your app to Azure or another host. Learn more

benday-webapp - Web Deploy

Site URL	http://benday-webapp.azur...
Resource Group	rg-benday-webapp
Configuration	Release
Troubleshooting Info	See Guide

14. A browser window will open that will take you to your **{{App Service URL}}**.
15. You'll be prompted to log in using your Microsoft Account (MSA) credentials. Log in.

16. You'll be prompted with a permissions dialog. Click **Yes**.


Microsoft

benday@live.com

Let this app access your info?

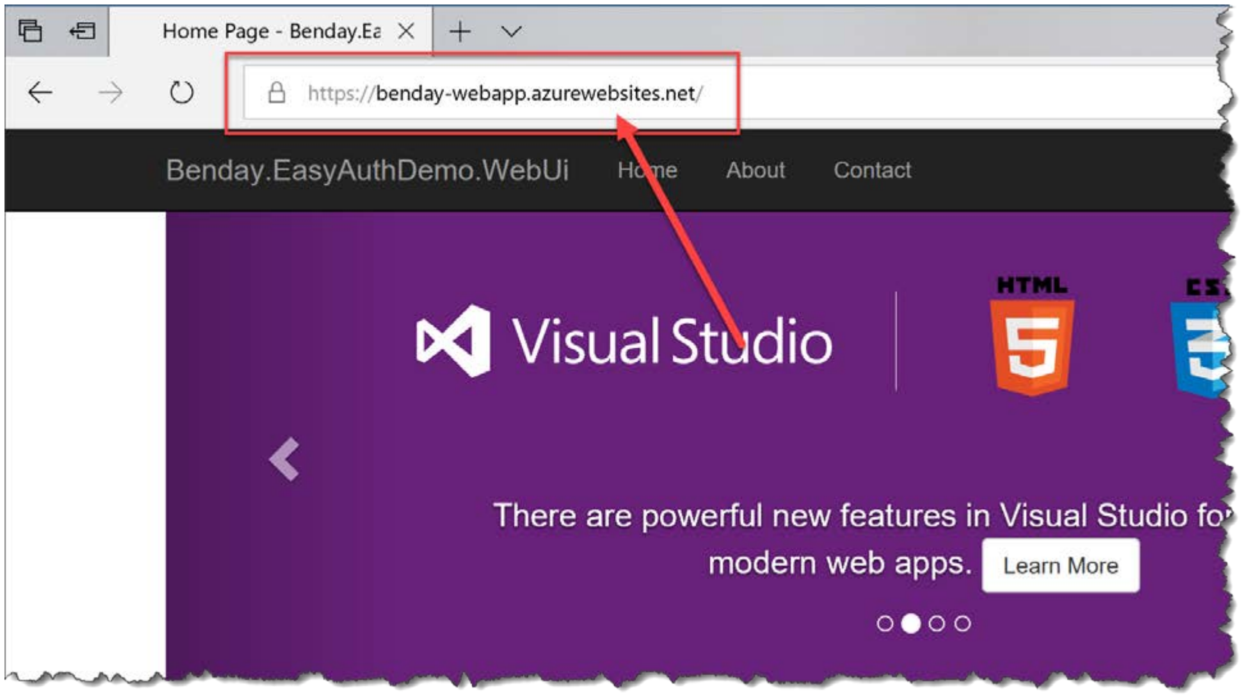
benday-webapp.azurewebsites.net

benday-webapp needs your permission to:

View your profile info and contact list
benday-webapp will be able to see your profile info, including your name, gender, display picture, contacts, and friends.

You can change these application permissions at any time in your account settings.

17. You should now see the sample app running in Azure at your **{{App Service URL}}**.


Home Page - Benday.Ea
https://benday-webapp.azurewebsites.net/
Benday.EasyAuthDemo.WebUi
Home
About
Contact
Visual Studio
HTML
There are powerful new features in Visual Studio fo
modern web apps.
Learn More

18. Click around in the application.

You've deployed your application and it's now secured. Before anyone accesses anything in your application, they need to be logged in. All that login logic is handled by your Azure Web App and by the Microsoft Account (MSA) security infrastructure.

Pretty cool, huh?

A little bit of configuration and you're done. You're using Azure "Easy Auth".