

Developing and
Deploying
Intelligent Chat
Bots



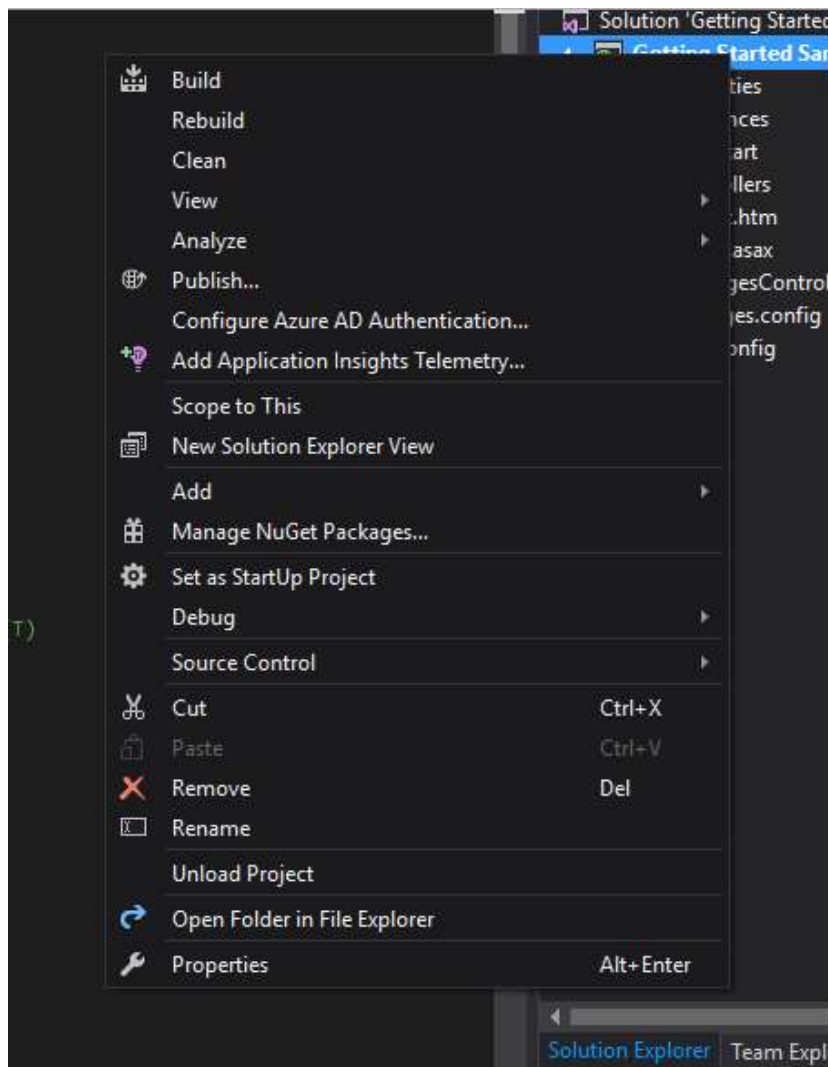
Registering and Publishing Bots
Lab

1. Publishing your Bot Application to Microsoft Azure

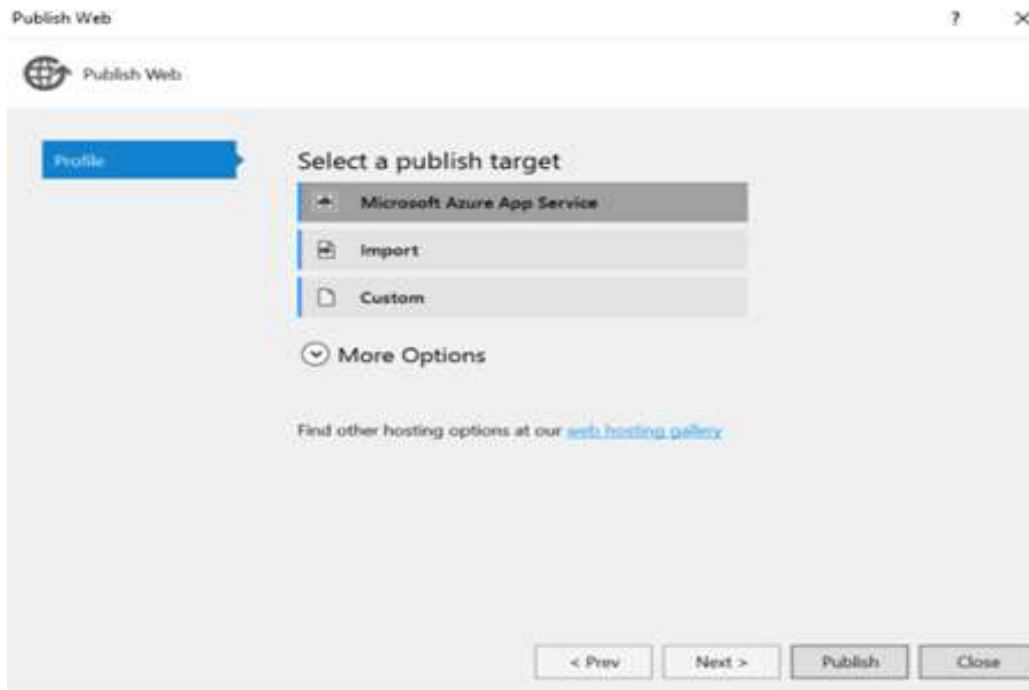
To publish your Bot Application you will need a Microsoft Azure subscription. You can get a free trial from here: azure.microsoft.com/en-us/

Right click on the project and choose "Publish", and then your appropriate Azure subscription information. By default, the bot should be published as an Microsoft Azure App Service. When publishing, keep track of the URL you chose because we'll need it to update the Bot Framework registration endpoint.

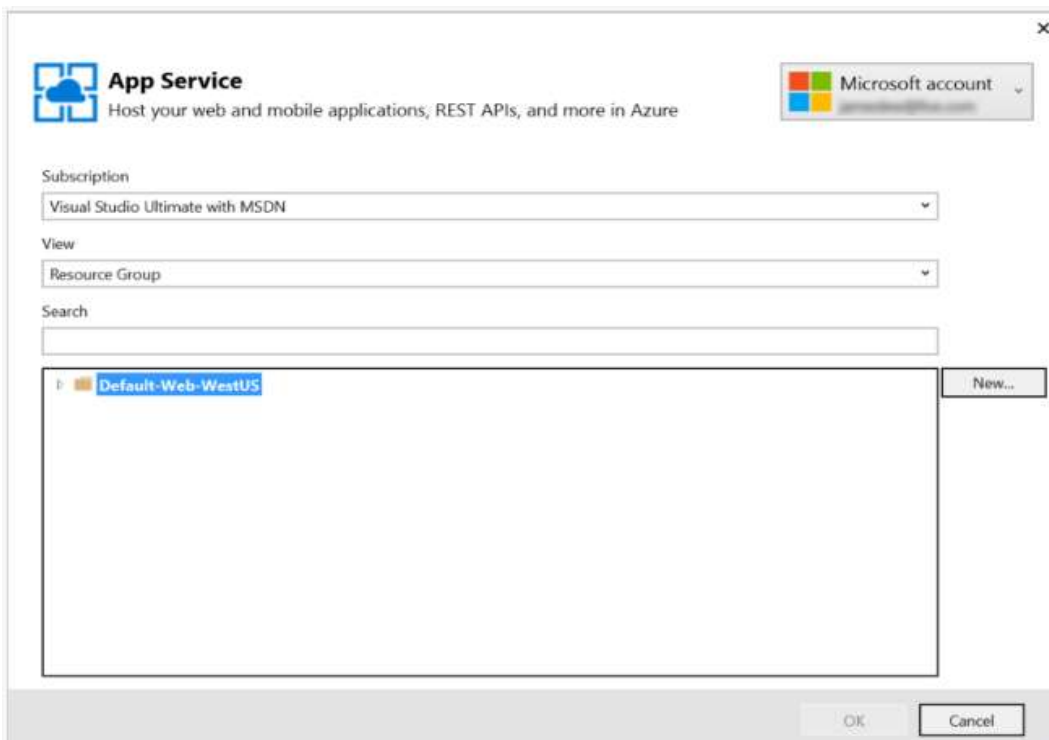
In Visual Studio, right click on the project in Solution Explorer and select "Publish"



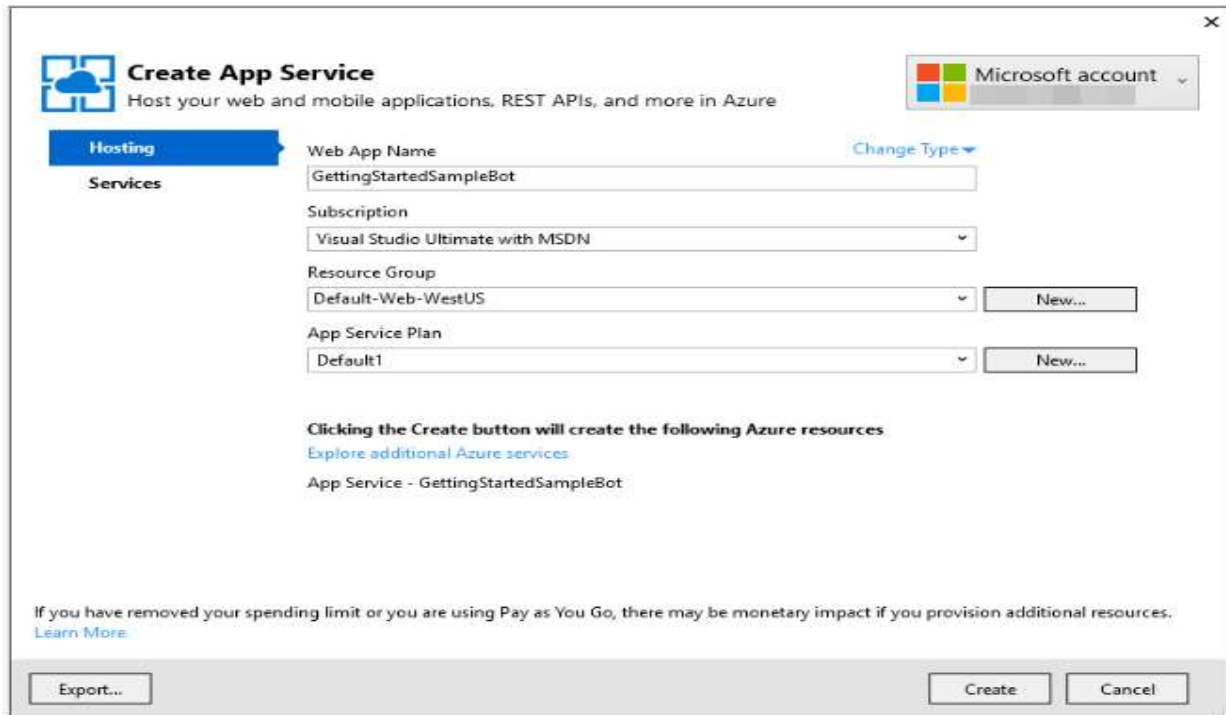
The Publish wizard will start. Select "Microsoft Azure App Service" as your project type.



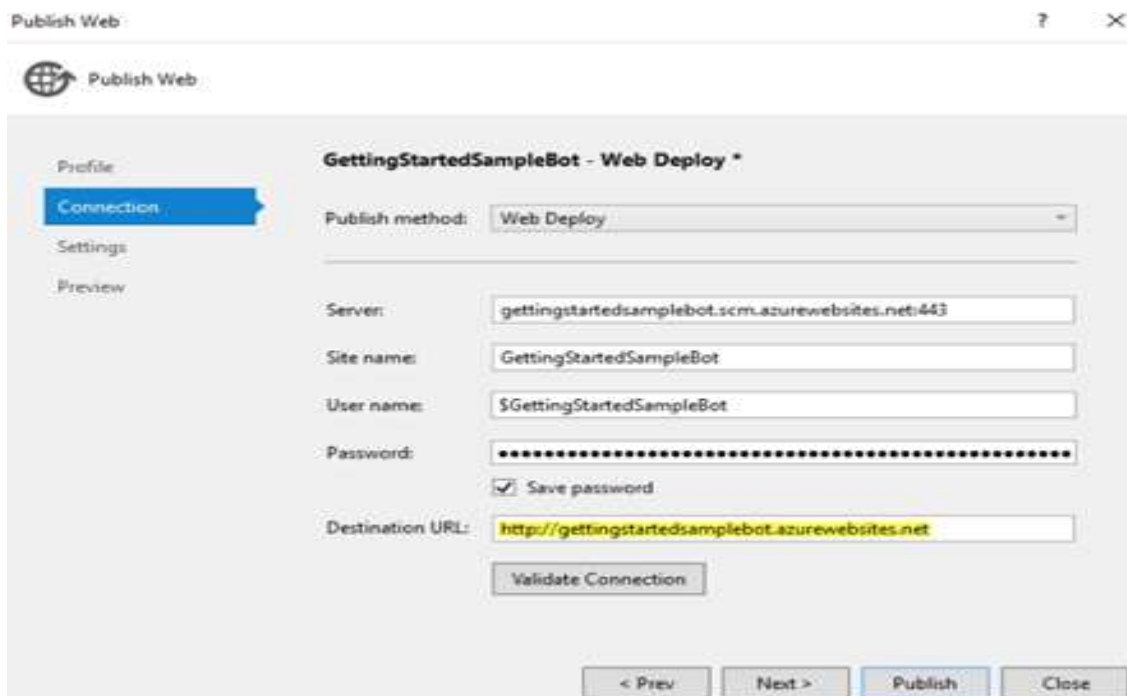
The next step in the Azure App Service publishing process is to create your App Service. Click on "New..." on the right side of the dialog to create the App Service.



The Create App Service dialog will be displayed, fill in the details as appropriate. Make sure to choose "Web App" from the Change Type drop down in the top right.

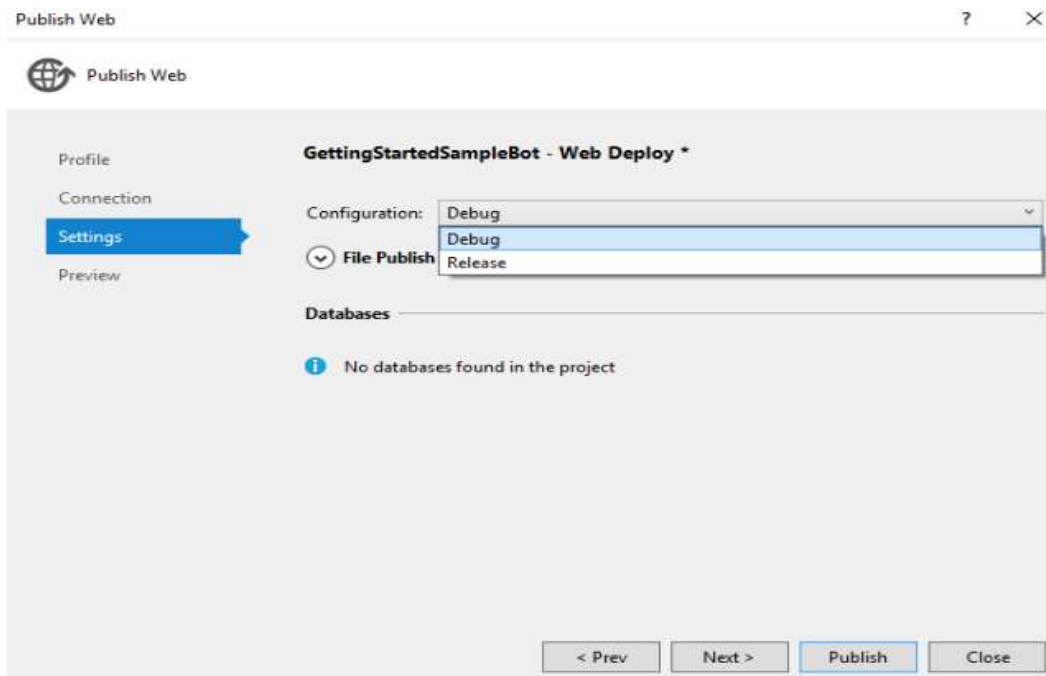


The "Create App Service" dialog box is shown. It features a Microsoft account icon in the top right. The "Hosting" tab is selected. The "Web App Name" field contains "GettingStartedSampleBot". The "Subscription" dropdown is set to "Visual Studio Ultimate with MSDN". The "Resource Group" dropdown is set to "Default-Web-WestUS", with a "New..." button next to it. The "App Service Plan" dropdown is set to "Default1", also with a "New..." button next to it. A "Change Type" link is visible. Below the fields, a message states: "Clicking the Create button will create the following Azure resources" followed by a link to "Explore additional Azure services" and the text "App Service - GettingStartedSampleBot". At the bottom, there is a warning about spending limits and a "Learn More" link. The bottom bar contains "Export...", "Create", and "Cancel" buttons.

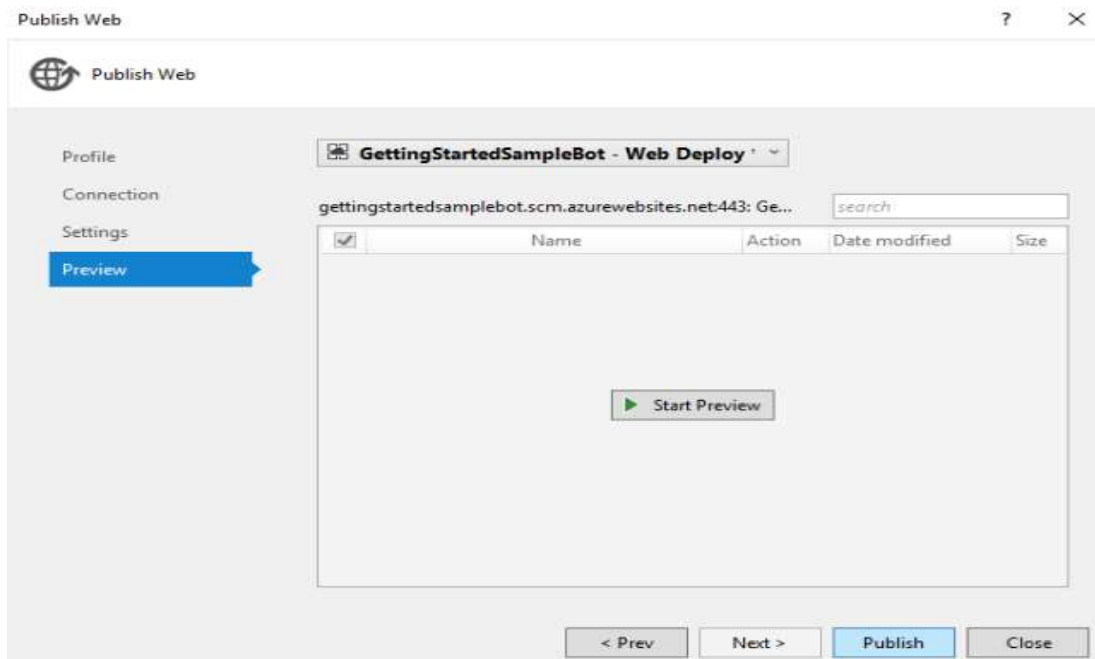


The "Publish Web" dialog box is shown. It has a "Publish Web" title bar. The "Connection" tab is selected. The "Publish method" is set to "Web Deploy". The "Server" field contains "gettingstartedsamplebot.scm.azurewebsites.net:443". The "Site name" field contains "GettingStartedSampleBot". The "User name" field contains "\$GettingStartedSampleBot". The "Password" field is masked with dots, and the "Save password" checkbox is checked. The "Destination URL" field contains "http://gettingstartedsamplebot.azurewebsites.net". A "Validate Connection" button is present. At the bottom, there are "< Prev", "Next >", "Publish", and "Close" buttons.

After returning to the Publish Web wizard, copy the destination URL to the clipboard. You will need it for the messaging endpoint during Bot registration (Section 2). Hit "Validate Connection" to ensure the configuration is good, and if all goes well, click "Next".



By default your Bot will be published in a Release configuration. If you want to debug your Bot, change Configuration to Debug. Regardless, from here you'll hit "Publish" and your Bot will be published to Azure.




You will see a number of messages displayed in the Visual Studio 2015 "Output" window. Once publishing is complete you will also see the web page for your Bot Application displayed in your browser.

2. Bot Registration

Sign in at <https://dev.botframework.com/bots> and click on the "Register a bot" tab. You will see the below page.

Tell us about your bot

Bot profile



Icon
[Upload custom icon](#)
30K max, png only

Name: * [?](#)

Bot handle: * [?](#)

Description: * [?](#)

Configuration

Messaging endpoint:

Register your bot with Microsoft to generate a new App ID and password
[Create Microsoft App ID and password](#)

Paste your app ID below to continue

Admin

Owners: [?](#)

AppInsights instrumentation key: [?](#)

☐ I agree to the [Terms of Use](#), [Privacy Statement](#), and [Code of Conduct](#) for the Microsoft Bot Framework (Preview).

[Register](#) [Cancel](#)

Unique Name Displayed in Bot Directory (35 character limit)

Used in the URL for your bot (alphanumeric and underscore only)

First 46 characters displayed on your bot card in Bot Directory

Add "/api/messages" to your URL from Azure Web App. Ensure https is used and not http

Refer to Section 3

3. Configuration

Once your registration is created, Microsoft Bot Framework will take you through generating your MicrosoftAppId and MicrosoftAppPassword. These are used to generate your MicrosoftAppPassword, be sure to record it somewhere as you won't be able to see it again.

Configuration

Messaging endpoint ?

* Microsoft App ID ?

Generate Microsoft App ID and password

Change the following keys in the web.config file to match the ones generated when you saved your registration, and you're ready to build.

```
<? xml version = "1.0" encoding = "utf-8" ?>

<!--

For more information on how to configure your ASP.NET application, please visit
http://go.microsoft.com/fwlink/?LinkId=301879

-->

< configuration >
< appSettings >
    <!--update these with your appid and one of your appsecret keys-->
    < add key = "MicrosoftAppId" value = "[GUID]" />
    < add key = "MicrosoftAppPassword" value = "[PASSWORD]" />
</ appSettings >
```

4. Testing the connection to your bot

In the developer dashboard of your Bot is a test chat window that you can use to interact with your Bot without further configuration, and verify that the Bot Framework can communicate with your Bot's web service.

Note that the first request after your Bot starts up can take 10 - 15 s as Azure starts up the web service for the first time. Subsequent requests will be quick. This simple viewer will let you see the JSON object returned by your Bot.

Test connection to Bot framework

Test

Endpoint authorization succeeded.