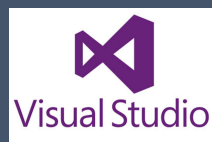
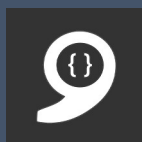




# BOT Work Shop

Eric Hullander



**Microsoft Partner**  
Gold Data Analytics  
Gold Cloud Platform



# Tools You Will Need

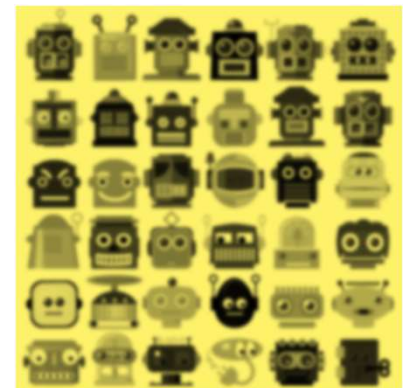
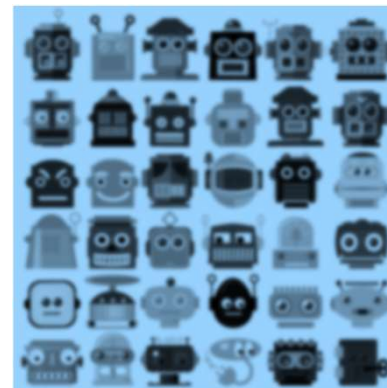
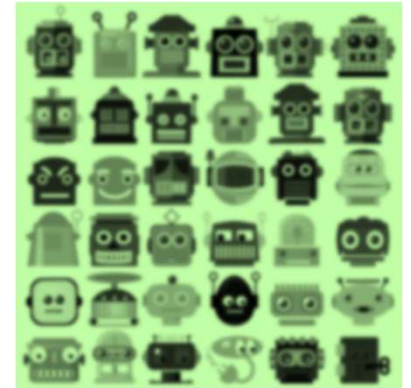
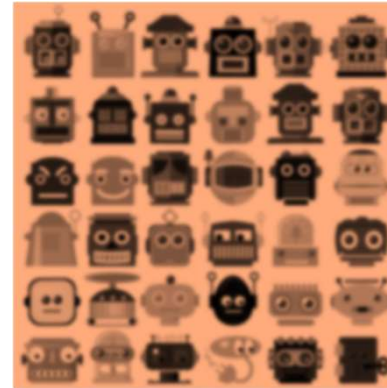
<https://docs.botframework.com/en-us/>

Required:

- Microsoft Account
- [Visual Studio](#) 2015 or higher
- Microsoft.Bot.Builder ("Manage NuGet Packages" in VS)
- [Bot emulator](#)
- Web Browser
- [Skype](#), etc.

Optional:

- SQL Management Studio
- Excel
- Notepad
- [PowerBI](#)



# Bot Framework

- [Sign into Azure](#)
- <https://dev.botframework.com/>
- Click Register a Bot
- Fill in
  - Name
  - Bot Handle
- Click Create Microsoft App ID and password
- Will leave “Messaging Endpoint” Blank

Bot Framework  
My bots Register a bot Documentation Bot Directory Blog

Tell us about your bot

Bot profile

Icon  
Upload custom icon  
300 max, any ratio

Name \*  
Bot handle \*  
Type in your Bot handle  
Description \*  
Description

Configuration

Messaging endpoint  
https URL

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  
Microsoft App ID from the Microsoft Apps registration portal

Admin

Owners  
eric.hullander@outlook.com

Instrumentation key  
Instrumentation key (Azure App Insights key)

By clicking Register, you agree to the Privacy statement, Terms of use, and Code of conduct.

Register Cancel

# Bot Framework

- Copy the App ID to notepad for future use

## Generate App ID and password

App name

App ID

[Generate an app password to continue](#)

☐ English (United States)

# Bot Framework AppID and Password

- Copy the App ID to notepad for future use
- Click “Generate an app password...”
- Copy the Password to notepad for future use
- Click “OK”
- “Click Finish and go to Bot Framework”

App name: EricHullandertest

App ID: c6851ae8-631c-4911-ae11-44a236edc3ad

App pw: L4wqs7mXeAnz7OGeCXWxFTm

Generate App ID and password

App name  
EricHullandertest

App ID  
c6851ae8-631c-4911-ae11-44a236edc3ad

[Generate an app password to continue](#)

☐ English (United States)

New password generated

Copy this password now, this is the only time when it will be displayed. Please store it securely and paste it into your bot configuration file.

L4wqs7mXeAnz7OGeCXWxFTm

[Ok](#)

Generate App ID and password

App name  
EricHullandertest

App ID  
c6851ae8-631c-4911-ae11-44a236edc3ad

Password [Learn More](#)  
L4w\*\*\*\*\*

[Finish and go back to Bot Framework](#)

☐ English (United States)

# Bot Framework Registration

- Add the description if you haven't
- Leave endpoint blank for now
- Click if you agree to the T&Cs
- Click "Register"
- Click "OK" on "Bot created" menu
- You have just taken your first step into a much larger world...

Option: You may add an app icon now or later if you like...

The screenshot shows the 'Bot Framework' registration interface. The top navigation bar includes links for 'Bot Framework', 'My bots', 'Register a bot', 'Documentation', 'Bot Directory', and 'Blog'. The main heading is 'Tell us about your bot'.

The 'Bot profile' section contains an 'icon' placeholder with the text 'Upload custom icon' and 'JPG, PNG, GIF, max 100KB'. Below this are input fields for 'Name' (containing 'Erichullandertest'), 'Bot handle' (containing 'erichullandertest'), and 'Description' (containing 'my test bot'). A red asterisk indicates that the description is a required field.

The 'Configuration' section includes a 'Messaging endpoint' field with 'https://URL'. Below this, it instructs the user to 'Register your bot with Microsoft to generate a new App ID and password' and provides a 'Manage Microsoft App ID and password' button. A 'Paste your app ID below to continue' field is also present, marked as a required field.

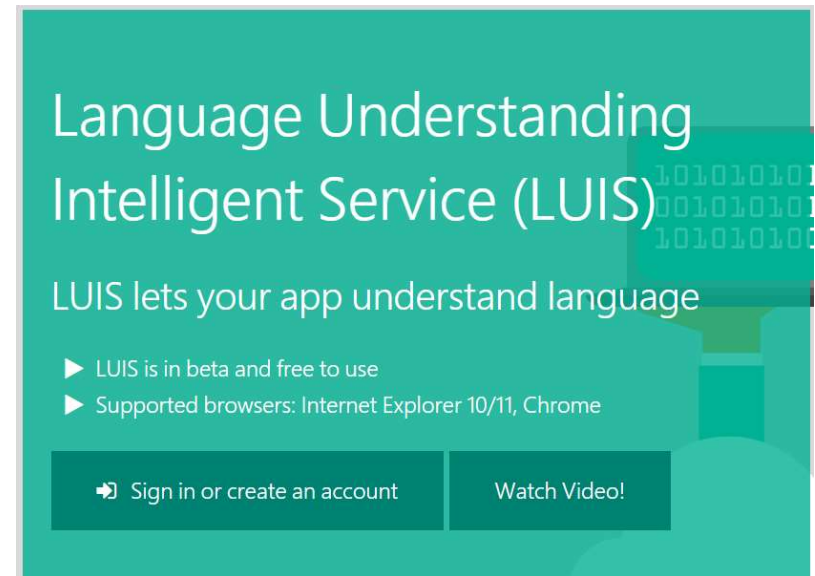
The 'Admin' section contains an 'Owners' field with 'erichullander@outlook.com', an 'Instrumentation key' field, and a note that the instrumentation key is for 'Azure App Insights key'.

At the bottom, a checkbox states 'By clicking Register, you agree to the Privacy statement, Terms of use, and Code of conduct'. There are 'Register' and 'Cancel' buttons.

A 'Bot created' confirmation dialog box is overlaid on the right side of the form, featuring an 'OK' button.

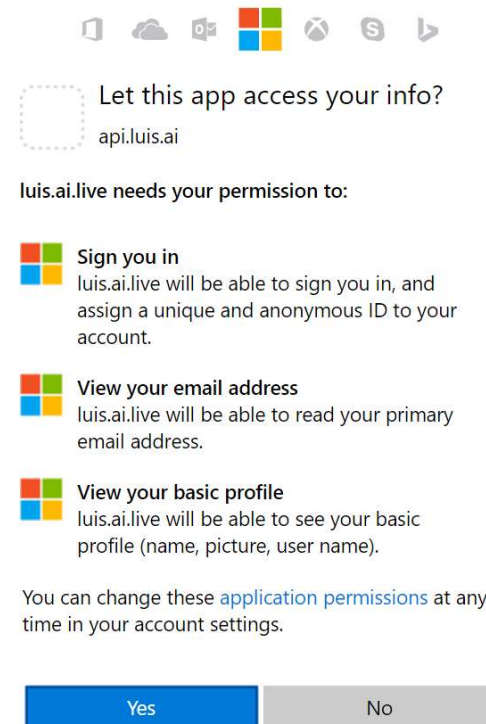
# LUIS

- [Sign into Azure](#)
- <https://www.luis.ai/>
- Sign in with your Microsoft account



# LUIS

- [Sign into Azure](#)
- <https://www.luis.ai/>
- Sign in with your Microsoft account
- “Yes”

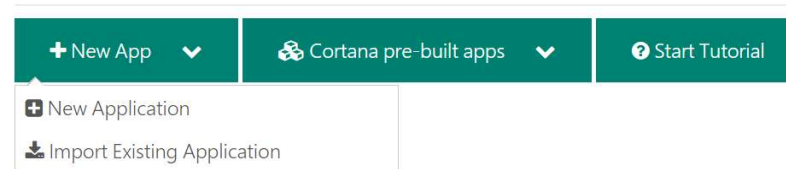




# LUIS

## Building a LUIS Model

- New App
- Choose “Import...”



My Applications

A LUIS Application (model) takes examples of many phrases (utterances) that mean the same thing and maps them to a single intent.

A LUIS Application can have up to 40 unique intents

Given a new utterance, LUIS predicts which of the 40 intents is the best match

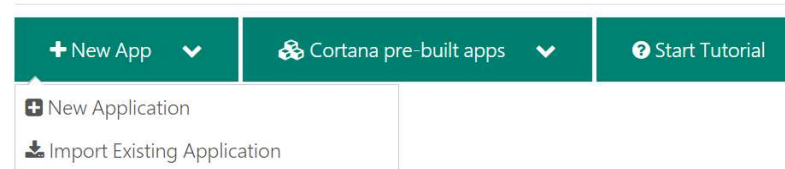
Let's get started

Build [a new language understanding application...](#)

# LUIS

## Building a LUIS Model

- New App
- Choose “Import...”



My Applications

Further, entities (nouns, verbs, adjectives) can be inferred from within an utterance

Example:

I want to buy a **house**.

I want to buy a **car**.

Might map to the intent “purchase”

The entity could be “house” or “car”

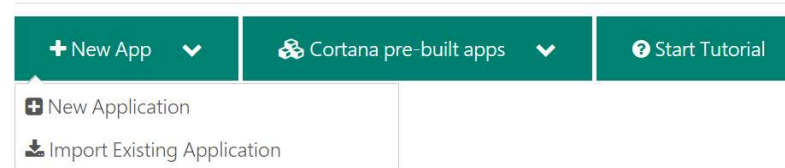
Let's get started

[Build a new language understanding application...](#)

# LUIS

## Building a LUIS Model

- New App
- Choose “Import...”



My Applications

To work best, LUIS needs many examples to learn from

Fortunately, if you’ve already trained LUIS before, and your application will be using the same set of intents and entities you can recycle an existing model and build on it

Let's get started

[Build a new language understanding application...](#)

# LUIS

## Building a LUIS Model

- Once a model has been trained, we can export it as a JSON file
- Click “Choose File” and Navigate to the JSON file provided
- Click “Import”



HotelExampleBot.json

### Import an existing application

Choose an application JSON file to import your application. Please note that importing an application may take several minutes. You will automatically be directed to your new app when it is successfully imported.

Choose File and enter (optional) imported app name:

HotelExampleBot.json

# LUIS JSON File

- Within the JSON file you will find lists of intents and later, text and entities that map to that intent
- You may edit this JSON directly and upload it, or use the interface



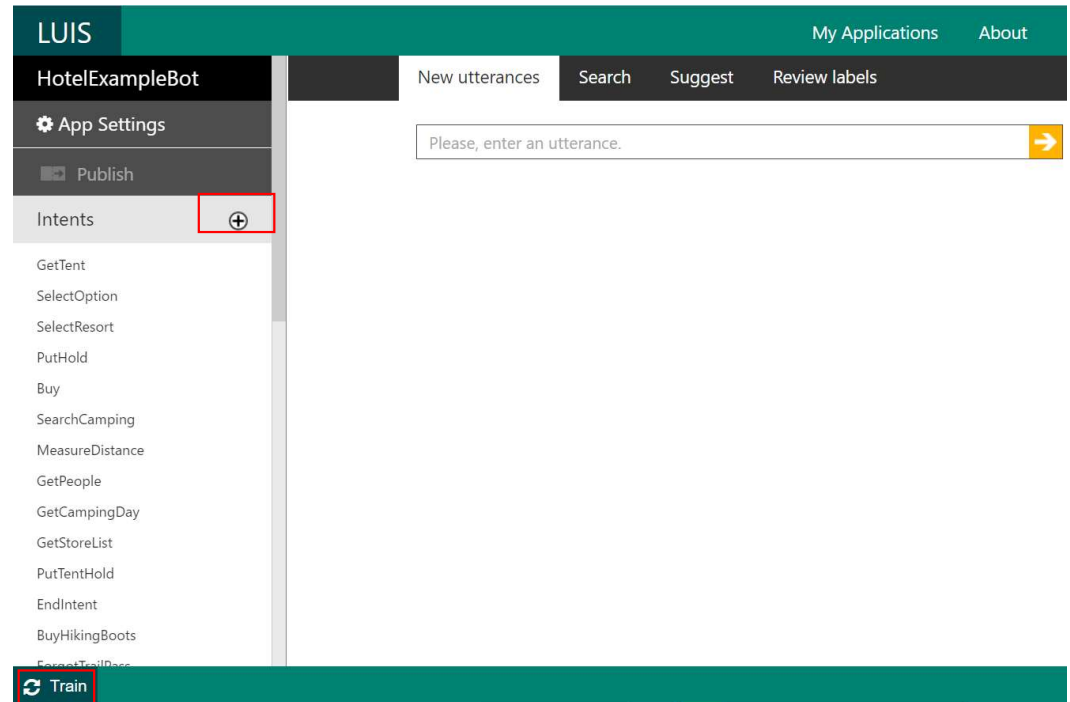
HotelExampleBot.json

```
{
  "luís_schema_version": "1.3.0",
  "name": "HotelExampleBot",
  "desc": "",
  "culture": "en-us",
  "intents": [
    {
      "name": "Buy"
    },
    {
      "name": "BuyHikingBoots"
    },
    {
      "name": "EndIntent"
    },
    {
      "name": "FindHikes"
    },
    {
      "name": "ForgotTrailPass"
    },
    {
      "text": "camp sites",
      "intent": "SearchCamping",
      "entities": []
    },
    {
      "text": "camping",
      "intent": "SearchCamping",
      "entities": []
    },
    {
      "text": "any hikes",
      "intent": "FindHikes",
      "entities": []
    },
    {
      "text": "do they have nice hikes?",
      "intent": "FindHikes",
      "entities": []
    }
  ]
}
```

# LUIS

## Editing

- Once Loaded we see this screen
- The list on the left are the “intents” this model understands
- If we click (+), we can add new intents to the model
- Even though we have imported the file, we still need to train the algorithm to recognize utterances
- Click “Train”



# LUIS Publishing

- Training can take some time. Click “Publish”
- Click “Publish web Service”

The screenshot displays the LUIS web interface. The top navigation bar includes 'LUIS', 'My Applications', and 'About'. Below this, the application name 'HotelExampleBot' is shown, along with tabs for 'New utterances', 'Search', 'Suggest', and 'Review labels'. A search input field with the placeholder text 'Please, enter an utterance.' and a submit button is present. On the left sidebar, the 'Publish' button is highlighted with a red rectangle. Below the sidebar, a list of intents is visible: GetTent, SelectOption, SelectResort, PutHold, Buy, SearchCamping, MeasureDistance, and GetPeople. A modal dialog titled 'HTTP service' is open, showing the text 'Publish Current Application to URL for access via HTTP' and 'Status: service not published'. The 'Publish web service' button in this dialog is also highlighted with a red rectangle. At the bottom of the dialog, a note states: 'Note: To enable bot integration, enable action fulfillment in one of your intents.' followed by a checkbox labeled 'Enable Action Binding using Microsoft Bot Framework'.

# LUIS Publishing

- Training can take some time. Click “Publish”
- Click “Publish web Service”

The screenshot displays the LUIS web interface. The top navigation bar includes 'LUIS', 'My Applications', and 'About'. Below this, the application name 'HotelExampleBot' is shown, along with tabs for 'New utterances', 'Search', 'Suggest', and 'Review labels'. A search input field with the placeholder text 'Please, enter an utterance.' and a submit button is present. The left sidebar contains a list of options: 'App Settings', 'Publish', and 'Intents'. The 'Publish' option is highlighted with a red rectangle. Below the sidebar, a list of intents is visible: 'GetTent', 'SelectOption', 'SelectResort', 'PutHold', 'Buy', 'SearchCamping', 'MeasureDistance', and 'GetPeople'. A modal dialog titled 'HTTP service' is open, showing the text 'Publish Current Application to URL for access via HTTP' and 'Status: service not published'. The 'Publish web service' button in this dialog is highlighted with a red rectangle. Below the dialog, a note states: 'Note: To enable bot integration, enable action fulfillment in one of your intents.' followed by a checkbox labeled 'Enable Action Binding using Microsoft Bot Framework'.



# LUIS

## Testing

- You can type an utterance into “Query” to see how the model scores it. Given an utterance it will give each possible intent in the model a score
- Example
- Query: “find a trail”

```
{ "query": "find a trail", "topScoringIntent": { "intent": "GetTrailPass", "score": 0.9974881 },  
  "intents": [ { "intent": "GetTrailPass", "score": 0.9974881 },  
    { "intent": "None", "score": 0.5163145 },  
    { "intent": "ForgotTrailPass", "score": 0.0379541256 },  
    { "intent": "Buy", "score": 0.0100398948 },  
    { "intent": "PutTentHold", "score": 0.007901444 },  
    { "intent": "Help", "score": 0.00303268922 },  
    { "intent": "SelectResort", "score": 0.00238058716 },
```

### HTTP service

Publish Current Application to URL for access via HTTP

Status: Published on 1/24/2017, 2:24:59 PM

[Update published application](#)

Query:

Optional Parameters

☒ Verbose

☐ Enable Bing Spell Check For This Query [?](#)

URL: <https://westus.api.cognitive.microsoft.com/luis/v2.0/apps/02849bf2-da96-4840-825c-73d340943a4a?subscription-key=>

**Note: To enable bot integration, enable action fulfillment in one of your intents.**

☐ Enable Action Binding using Microsoft Bot Framework

[Download web service usage logs](#)

[Download logs](#)

# LUIS

## AppID and Key

- Click App Settings
- Copy the App Id to your notepad
- In the top banner, click on your account name

LUIS

My Applications About

HotelExampleBot

New utterances Search Suggest Review labels

Please, enter an utterance. →

App Settings

Basic Settings Auth Settings

General Info

App Id: 02849bf2-da96-4840-825c-73d340943a4a

App Name

HotelExampleBot Rename Revert

Endpoint Key Assigned to app:

Save a new endpoint key:

Type in a new key ... Add New Key

Click here to buy a key from Azure

Select endpoint key

Please select...

Bing Spell Check

Enable Bing Spell Check. Click here to subscribe for a Bing spell check key!

Train Last train completed: 1/24/2017, 2:09:37 PM

# LUIS

## AppID and Key

- In the top banner, click on your account name
- Click Subscription Keys
- Copy the API Key to your notepad

App name: EricHullandertest

App ID: c6851ae8-631c-4911-ae11-44a236edc3ad

App pw: L4wqs7mXeAnz7OGcXWxFTm

LUIS App Name: HotelExampleBot

LUIS ADD ID: 02849bf2-da96-4840-825c-73d340943a4a

LUIS Key: 22296c1f55054d48b514027a0534f7ec



### My Settings



#### Programmatic API Key



Enter new key [Need help getting your key?](#)

[Copy your key from the Azure Portal ...](#)

Azure subscription keys ([click here to buy a key from Azure](#))

Application Name

HotelExampleBot

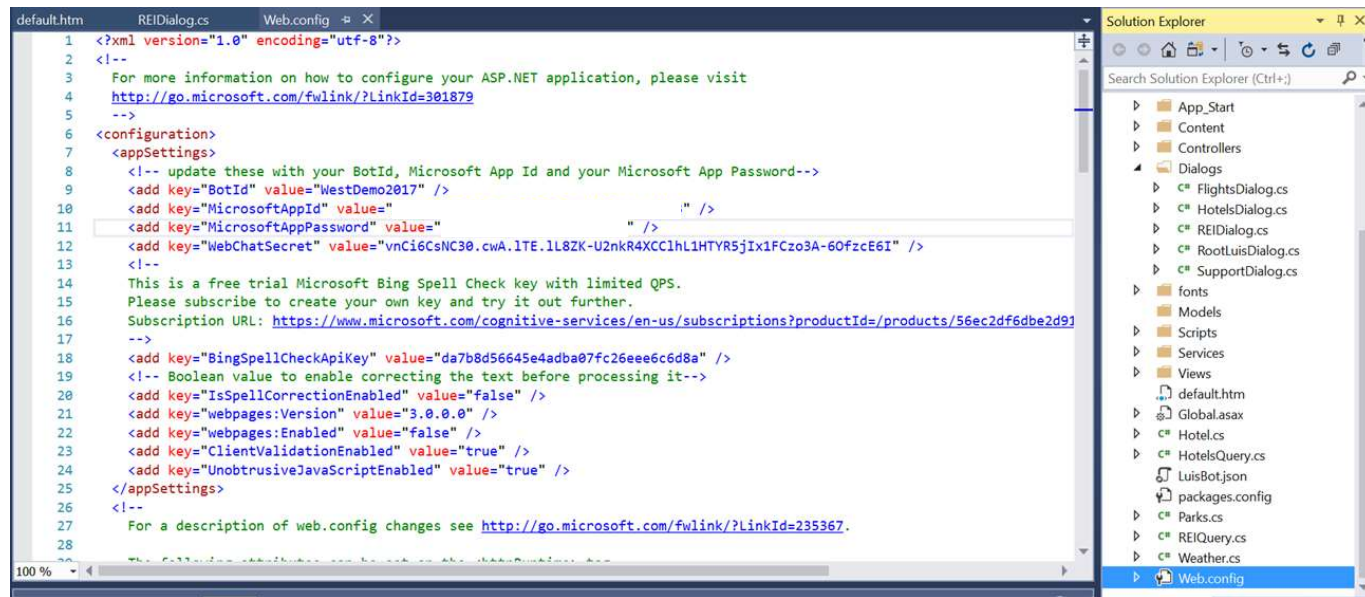
# Visual Studio

- Unzip and then navigate to the .sln file in the the zip file provided
- Open in Visual Studio
- This can be used as a template for future bot building and will provide a starting point for your exploration
- Some configuration is required

Name	Date modified	Type	Size
obj	1/24/2017 12:12 AM	File folder	
packages	11/28/2016 11:54 AM	File folder	
Properties	11/28/2016 11:55 AM	File folder	
Scripts	11/28/2016 11:55 AM	File folder	
Service References	11/28/2016 11:55 AM	File folder	
Services	11/28/2016 11:55 AM	File folder	
Views	11/28/2016 11:55 AM	File folder	
default.htm	1/24/2017 12:20 AM	Chrome HTML Docu...	2 KB
Global.asax	11/28/2016 11:54 AM	ASP.NET Server Appli...	1 KB
Global.asax.cs	11/28/2016 11:54 AM	Visual C# Source File	1 KB
Hotel.cs	11/28/2016 11:55 AM	Visual C# Source File	1 KB
HotelsQuery.cs	11/28/2016 11:54 AM	Visual C# Source File	1 KB
LuisBot.csproj	1/24/2017 12:12 AM	Visual C# Project file	16 KB
LuisBot.csproj.user	1/24/2017 12:12 AM	Per-User Project Opti...	2 KB
LuisBot.csproj.vspssc	11/28/2016 11:55 AM	Source Control Proje...	1 KB
LuisBot.json	11/28/2016 11:54 AM	JSON File	5 KB
LuisBot.sln	11/28/2016 11:54 AM	Visual Studio Solution	2 KB
packages.config	11/28/2016 11:54 AM	XML Configuration File	3 KB
Parks.cs	11/28/2016 11:54 AM	Visual C# Source File	1 KB
README.md	11/28/2016 11:54 AM	MD File	9 KB
REIQuery.cs	11/28/2016 11:54 AM	Visual C# Source File	1 KB
Weather.cs	11/28/2016 11:54 AM	Visual C# Source File	3 KB
Web.config	1/23/2017 11:41 PM	XML Configuration File	6 KB
Web.Debug.config	11/28/2016 11:54 AM	XML Configuration File	2 KB
Web.Release.config	11/28/2016 11:54 AM	XML Configuration File	2 KB

# Visual Studio

- Unzip and then navigate to the .sln file in the the zip file provided
- Open in Visual Studio
- Open Web.config
- Fill in from your notepad
  - BotId
  - MicrosoftAppId
  - MicrosoftAppPassword

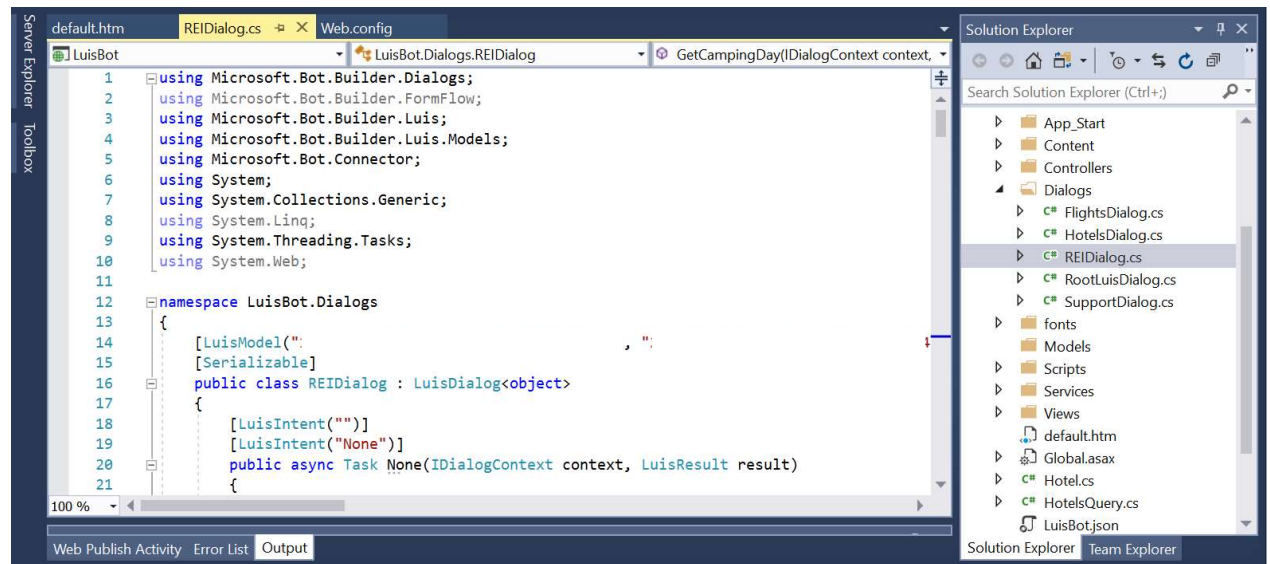


The screenshot shows the Visual Studio IDE with the `Web.config` file open in the editor. The file contains XML configuration for an ASP.NET application, including settings for BotId, MicrosoftAppId, MicrosoftAppPassword, and various API keys. The Solution Explorer on the right shows the project structure, including folders like `App_Start`, `Content`, `Controllers`, `Dialogs`, `fonts`, `Models`, `Scripts`, `Services`, `Views`, and files like `default.htm`, `Global.asax`, `Hotel.cs`, `HotelsQuery.cs`, `LuisBot.json`, `packages.config`, `Parks.cs`, `REIQuery.cs`, `Weather.cs`, and `Web.config`.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <!--
3   For more information on how to configure your ASP.NET application, please visit
4   http://go.microsoft.com/fwlink/?LinkId=301879
5   -->
6 <configuration>
7   <appSettings>
8     <!-- update these with your BotId, Microsoft App Id and your Microsoft App Password-->
9     <add key="BotId" value="WestDemo2017" />
10    <add key="MicrosoftAppId" value="" />
11    <add key="MicrosoftAppPassword" value="" />
12    <add key="WebChatSecret" value="vnci6CsNC30.cwA.1TE.1L8ZK-U2nKR4XCC1hL1HTYR5jIx1FCzo3A-60fzcE6I" />
13    <!--
14     This is a free trial Microsoft Bing Spell Check key with limited QPS.
15     Please subscribe to create your own key and try it out further.
16     Subscription URL: https://www.microsoft.com/cognitive-services/en-us/subscriptions?productId=/products/56ec2df6dbe2d91
17    -->
18    <add key="BingSpellCheckApiKey" value="da7b8d56645e4adba07fc26eee6c6d8a" />
19    <!-- Boolean value to enable correcting the text before processing it-->
20    <add key="IsSpellCorrectionEnabled" value="false" />
21    <add key="webpages:Version" value="3.0.0.0" />
22    <add key="webpages:Enabled" value="false" />
23    <add key="ClientValidationEnabled" value="true" />
24    <add key="UnobtrusiveJavaScriptEnabled" value="true" />
25  </appSettings>
26  <!--
27   For a description of web.config changes see http://go.microsoft.com/fwlink/?LinkId=235367.
28  -->
```

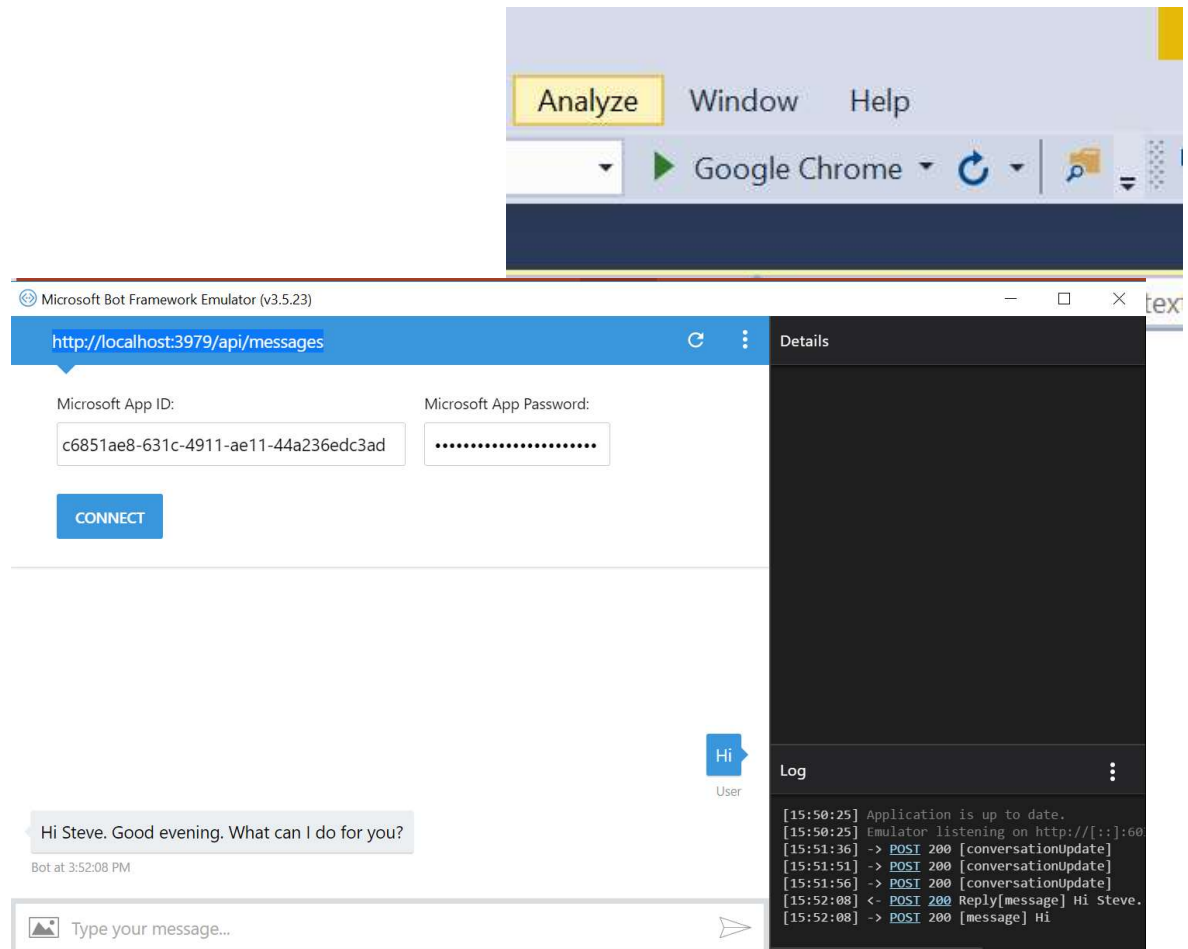
# Visual Studio

- Open REIDialog.cs
- Enter the App ID and Password



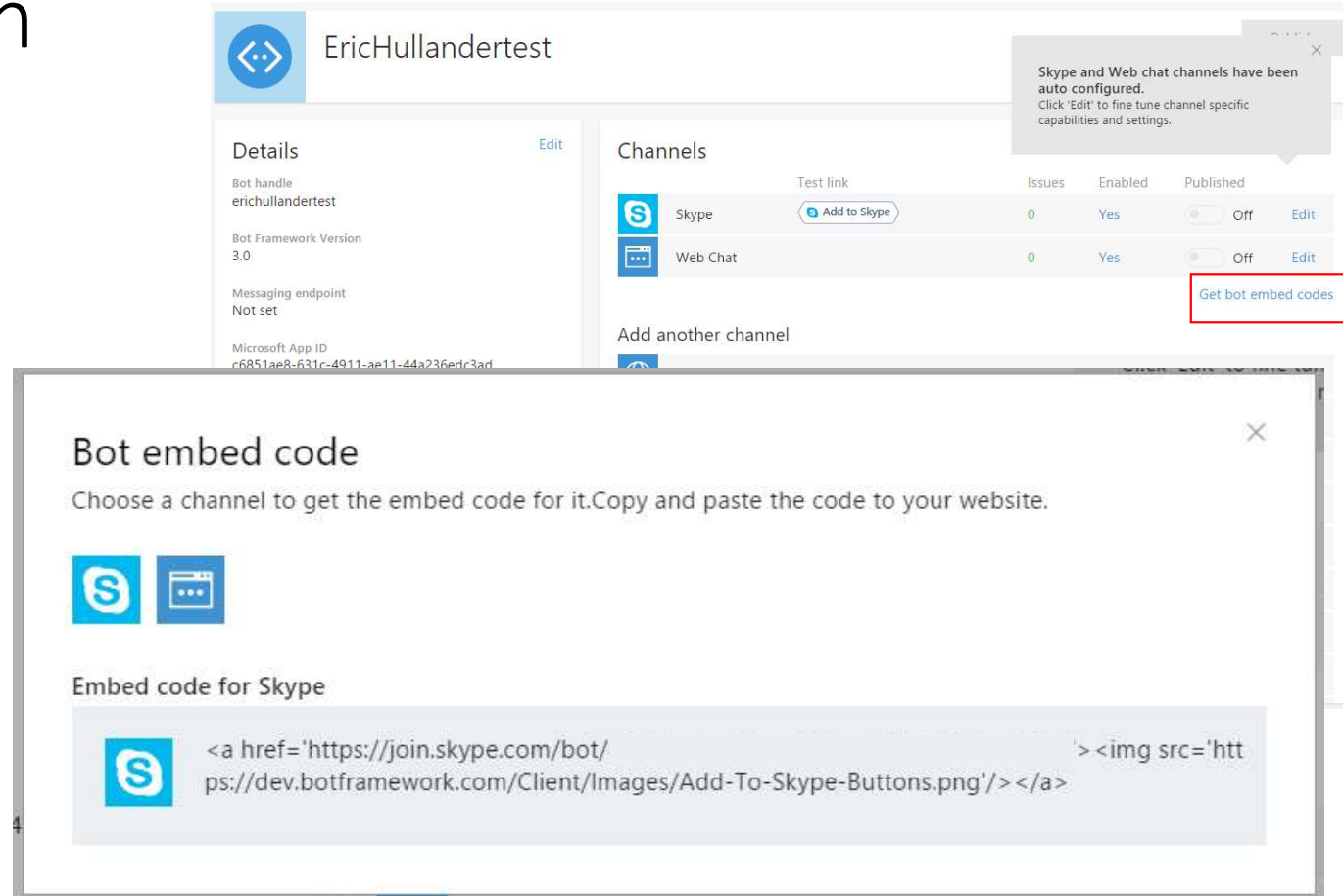
# Bot Emulation

- Before we publish we will test the bot locally using the bot emulator
- Click the Play Button to have the bot run on your local host
- Launch the Bot emulator
- Input the address of your local host
- Input your AppID and PW
- Let the emulator connect
- Say “Hi”



# Bot Publication Skype

- Return to <https://dev.botframework.com/bots?id=erichullandertest>
- Click the light blue Skype icon
- Click “Get bot embed codes”
- Copy the embed code to your notepad



EricHullandertest

Skype and Web chat channels have been auto configured. Click 'Edit' to fine tune channel specific capabilities and settings.

**Details** [Edit](#)



Bot handle  
erichullandertest

Bot Framework Version  
3.0

Messaging endpoint  
Not set

Microsoft App ID  
c6851ae8-631c-4911-ae11-44a236edc3ad

**Channels**



	Test link	Issues	Enabled	Published	
 Skype	<a href="#">Add to Skype</a>	0	Yes	Off	<a href="#">Edit</a>
 Web Chat		0	Yes	Off	<a href="#">Edit</a>

[Get bot embed codes](#)

Add another channel

**Bot embed code**

Choose a channel to get the embed code for it. Copy and paste the code to your website.

Embed code for Skype

```
<a href='https://join.skype.com/bot/ps://dev.botframework.com/Client/Images/Add-To-Skype-Buttons.png'><img src='htt
```



# Bot Publication Web

- Click the dark blue web icon
- “Click here to open the web...”

The screenshot shows the Microsoft Bot Framework portal for a bot named 'EricHullandertest'. The 'Details' tab on the left shows the bot handle 'erichullandertest' and framework version '3.0'. The 'Channels' tab on the right lists 'Skype' and 'Web Chat'. A red box highlights the 'Get bot embed codes' link. A modal window titled 'Bot embed code' is open, showing instructions to choose a channel and get the embed code. It includes icons for Skype and Web Chat, and a message stating 'Embed code for Web Chat has moved' with a link to the 'Web Chat configuration page'.

EricHullandertest

Skype and Web chat channels have been auto configured. Click 'Edit' to fine tune channel specific capabilities and settings.

Details

Bot handle  
erichullandertest

Bot Framework Version  
3.0

Messaging endpoint



Channels

	Test link	Issues	Enabled	Published	
Skype	<a href="#">Add to Skype</a>	0	Yes	Off	<a href="#">Edit</a>
Web Chat		0	Yes	Off	<a href="#">Edit</a>

[Get bot embed codes](#)

**Bot embed code**

Choose a channel to get the embed code for it. Copy and paste the code to your website.


Embed code for Web Chat has moved

[Click here to open the Web Chat configuration page >](#)

# Bot Publication Web

- Click “Add New Site”
- Give your site a name
  - “MySite”
- Click Done

Configure Web Chat



+ Add new site

How would you name your site?

Site name is for your reference and you can change it anytime.

Cancel Done

# Bot Publication Web

- Show
- Copy Secret to notepad
- Copy Embed Code to notepad
- Replace "YOUR\_SECRET\_HERE" in embed code with your secret

The screenshot shows the 'Bot Framework Web Chat' publication page for a site named 'EricHullanderTest'. The interface includes a sidebar with a '+ Add new site' button and the current site name. The main content area is divided into sections: 'Secret keys' with two input fields containing placeholder text and 'Show | Regenerate' links; 'Embed code' with a text area containing an iframe snippet and a 'Copy' link; and 'Preview' with a description of the preview feature and an 'Enable Preview' checkbox that is currently checked. The top right of the interface has a 'Disable' button and a trash icon.

+ Add new site

EricHullanderTest

EricHullanderTest

Secret keys

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

Show | Regenerate

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

Show | Regenerate

Embed code

<iframe src='https://webchat.botframework.com/embed/erichullander?test?s=YOUR\_SECRET\_HERE'></iframe>

Copy

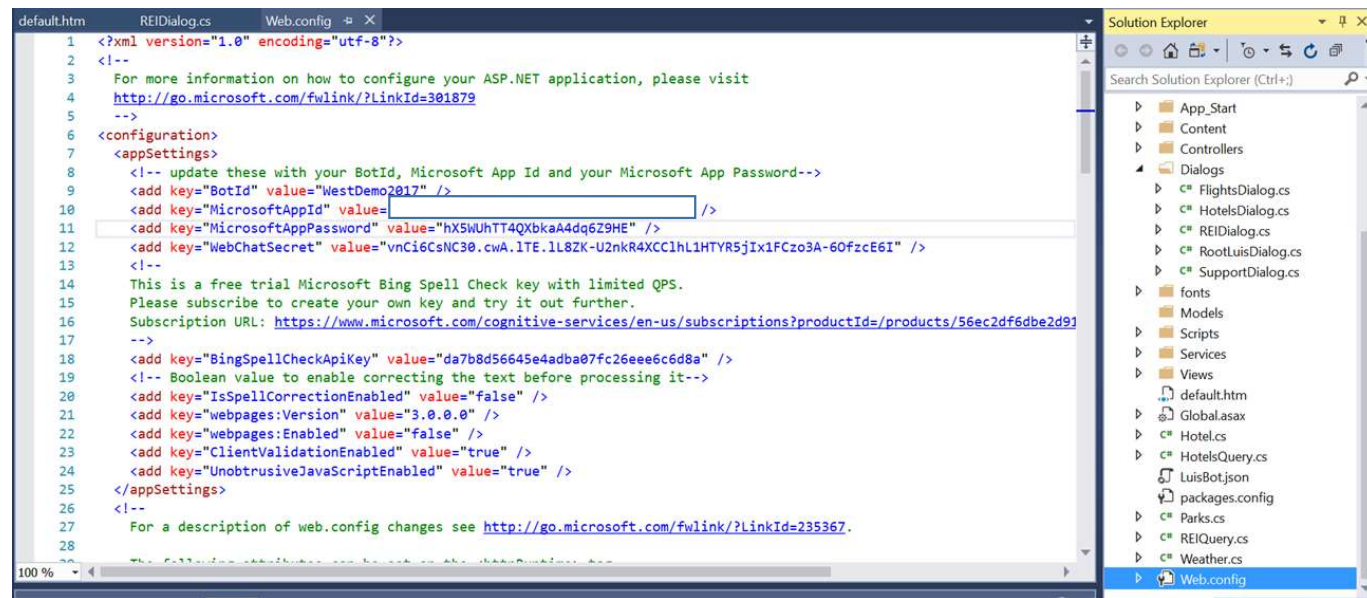
Preview

Enabling Web Chat Preview on this site will give users of your bot a preview version of the Bot Framework Web Chat control. The preview version includes new, beta features that have not yet been officially released.

☒ Enable Preview

# Visual Studio

- The web secret goes here too



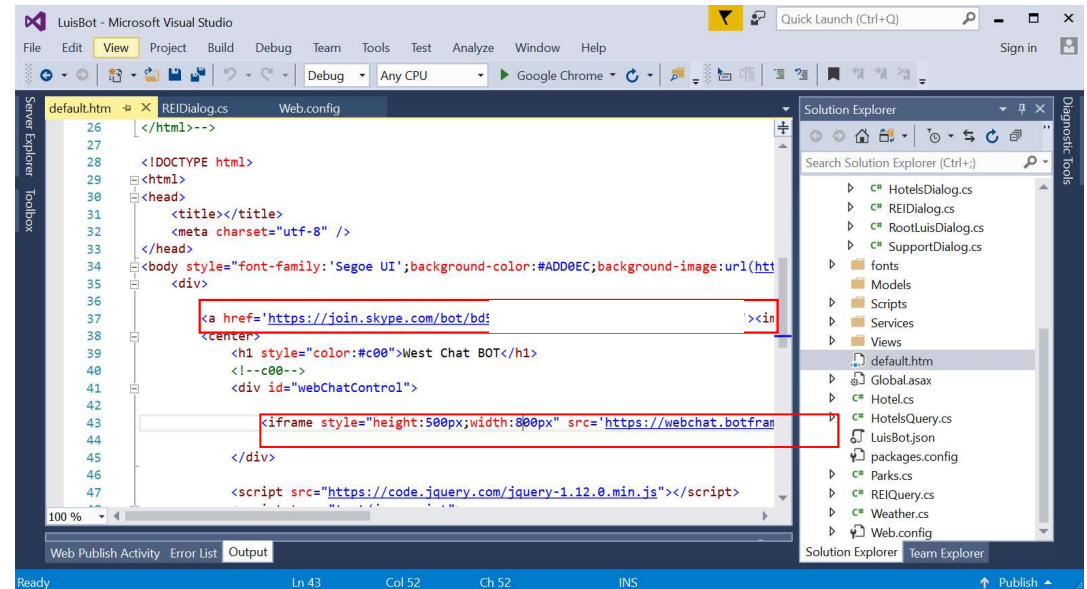
The screenshot shows the Visual Studio IDE with the `Web.config` file open. The file contains XML configuration for an ASP.NET application. The `<appSettings>` section is highlighted, showing several keys and values. The `MicrosoftAppId` value is being edited in a text box. The `WebChatSecret` value is also visible. The Solution Explorer on the right shows the project structure, including folders like `App_Start`, `Content`, `Controllers`, `Dialogs`, `fonts`, `Models`, `Scripts`, `Services`, `Views`, and files like `default.htm`, `Global.asax`, `Hotel.cs`, `HotelsQuery.cs`, `LuisBot.json`, `packages.config`, `Parks.cs`, `REIQuery.cs`, `Weather.cs`, and `Web.config`.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <!--
3   For more information on how to configure your ASP.NET application, please visit
4   http://go.microsoft.com/fwlink/?LinkId=301879
5   -->
6 <configuration>
7   <appSettings>
8     <!-- update these with your BotId, Microsoft App Id and your Microsoft App Password-->
9     <add key="BotId" value="WestDemo2017" />
10    <add key="MicrosoftAppId" value=" " />
11    <add key="MicrosoftAppPassword" value="hX5mUhtT4QXbkaA4dg6Z9HE" />
12    <add key="WebChatSecret" value="vnci6CsNC30.cwA.1TE.1L8ZK-U2nKR4XCC1hL1HTYR5jIx1FCzo3A-6OfzcE6I" />
13    <!--
14     This is a free trial Microsoft Bing Spell Check key with limited QPS.
15     Please subscribe to create your own key and try it out further.
16     Subscription URL: https://www.microsoft.com/cognitive-services/en-us/subscriptions?productId=/products/56ec2df6dbe2d91
17     -->
18    <add key="BingSpellCheckApiKey" value="da7b8d56645e4adba07fc26eee6c6d8a" />
19    <!-- Boolean value to enable correcting the text before processing it-->
20    <add key="IsSpellCorrectionEnabled" value="false" />
21    <add key="webpages:Version" value="3.0.0.0" />
22    <add key="webpages:Enabled" value="false" />
23    <add key="ClientValidationEnabled" value="true" />
24    <add key="UnobtrusiveJavaScriptEnabled" value="true" />
25  </appSettings>
26 <!--
27   For a description of web.config changes see http://go.microsoft.com/fwlink/?LinkId=235367.
28 -->
```

# Bot Publication Web

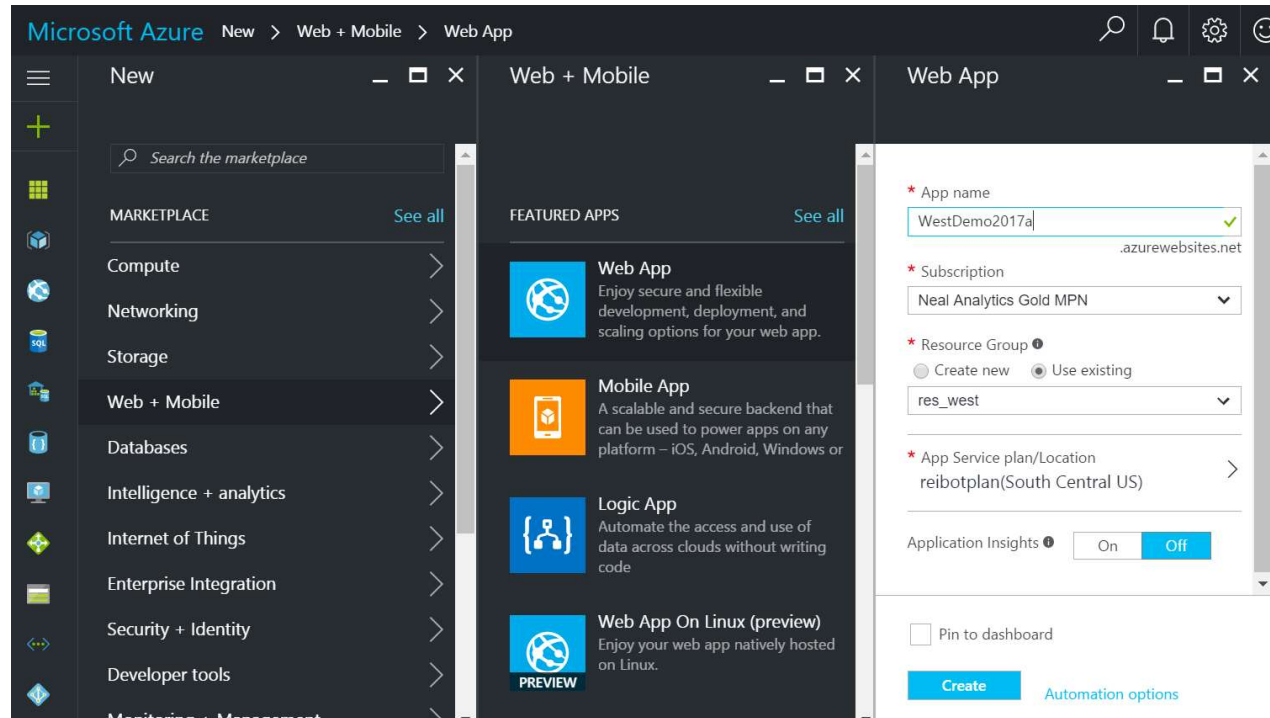
- Navigate to default.htm in Visual Studio
- Find the <a href> tag and replace with the <a href> tag for Skype in your notepad
- Find the <iframe> tag and replace it with the iframe tag in your notepad

This will enable Skype and the web interface for your bot



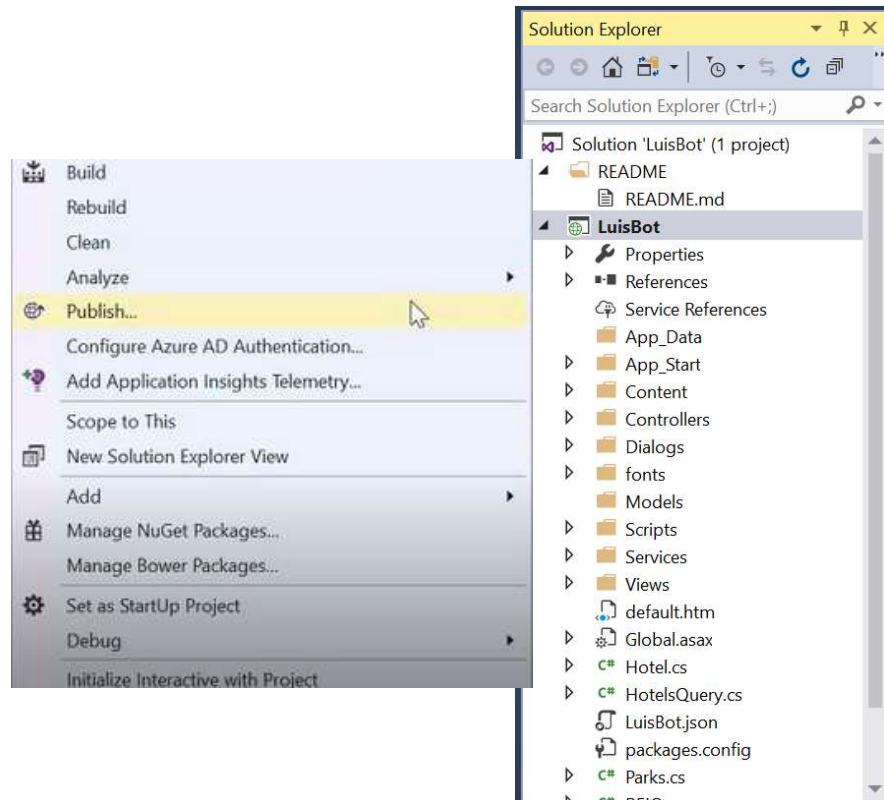
# Bot Publication Azure Portal

- <https://portal.azure.com/>
- Click +
- Click “Web + Mobile”
- Web App



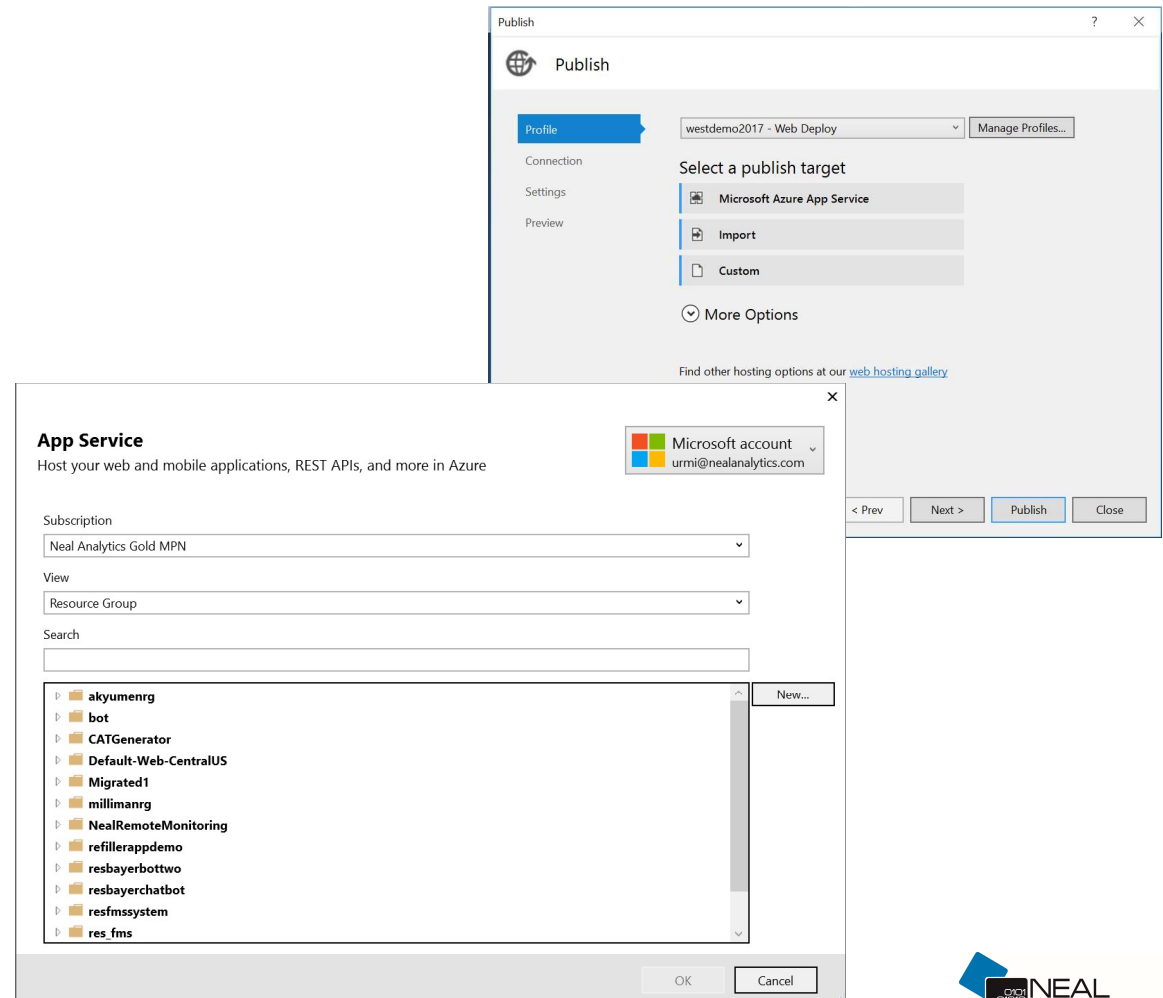
# Bot Publication Web

- Right Click on your solution
- Click “Publish”



# Bot Publication Web

- Click Profile
- Click Manage Profiles
- Remove any profiles (these are unique to the machine)
- Click “Microsoft Azure App Service”
- Select an account and Sign in
- Select the resource group you just created or selected

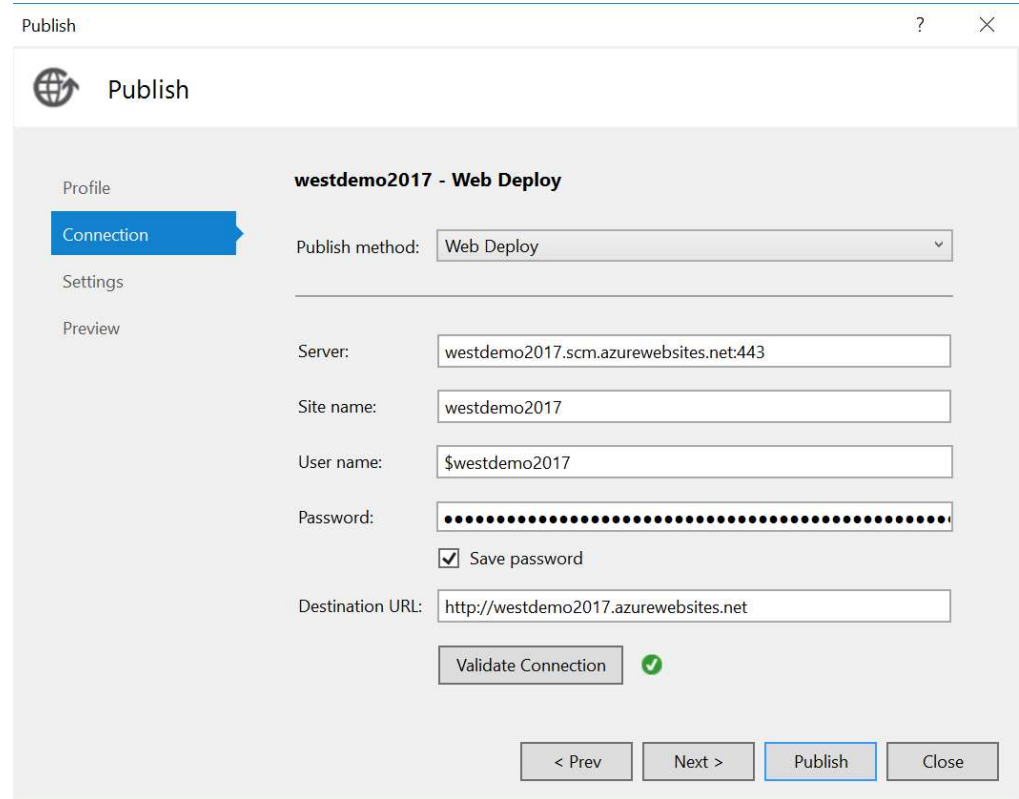




# Bot Publication Web

- Click “Validate Connection”
- Click “Publish”

Security Feature: You can only publish this app from this machine from now on!!!



The screenshot shows the 'Publish' dialog box in Visual Studio. The title bar says 'Publish'. On the left, there's a sidebar with 'Profile', 'Connection' (highlighted in blue), 'Settings', and 'Preview'. The main area is titled 'westdemo2017 - Web Deploy'. It contains the following fields and controls:

- Publish method:** A dropdown menu set to 'Web Deploy'.
- Server:** A text box containing 'westdemo2017.scm.azurewebsites.net:443'.
- Site name:** A text box containing 'westdemo2017'.
- User name:** A text box containing '\$westdemo2017'.
- Password:** A text box filled with dots, with a checkbox labeled 'Save password' checked below it.
- Destination URL:** A text box containing 'http://westdemo2017.azurewebsites.net'.
- Validate Connection:** A button with a green checkmark icon to its right.
- Navigation buttons:** At the bottom, there are four buttons: '< Prev', 'Next >', 'Publish' (highlighted in blue), and 'Close'.


# Bot is Live

- Click “Validate Connection”
- Click “Publish”

Security Feature: You can only publish this app from this machine from now on!!!

- Return to <https://dev.botframework.com/bots?id=erichullandertest>
- Click “Edit”
- Specify an endpoint

⚠ You cannot test or publish your bot until you specify a messaging endpoint.

 EricHullandertest

### Details

[Edit](#)

Bot handle  
erichullandertest

Bot Framework Version  
3.0

Messaging endpoint  
Not set

Microsoft App ID  
c6851ae8-631c-4911-ae11-44a236edc3ad

### Test connection to your bot

[Test](#)

### Configuration

Messaging endpoint:

Register your bot with Microsoft to generate a new App ID and password

[Manage Microsoft App ID and password](#)

Paste your app ID below to continue