

Developing and Deploying Intelligent Chat Bots

Objectives

Objectives

In certain scenarios such as testing, communication directly to your bot may be required. Communication between your bot and your own client application can be performed using Direct Line API. This hands-on lab introduces key concepts related to Direct Line API.

Setup

For this lab, you can use a published Bot to talk to via Direct Line API.

Download the sample from:

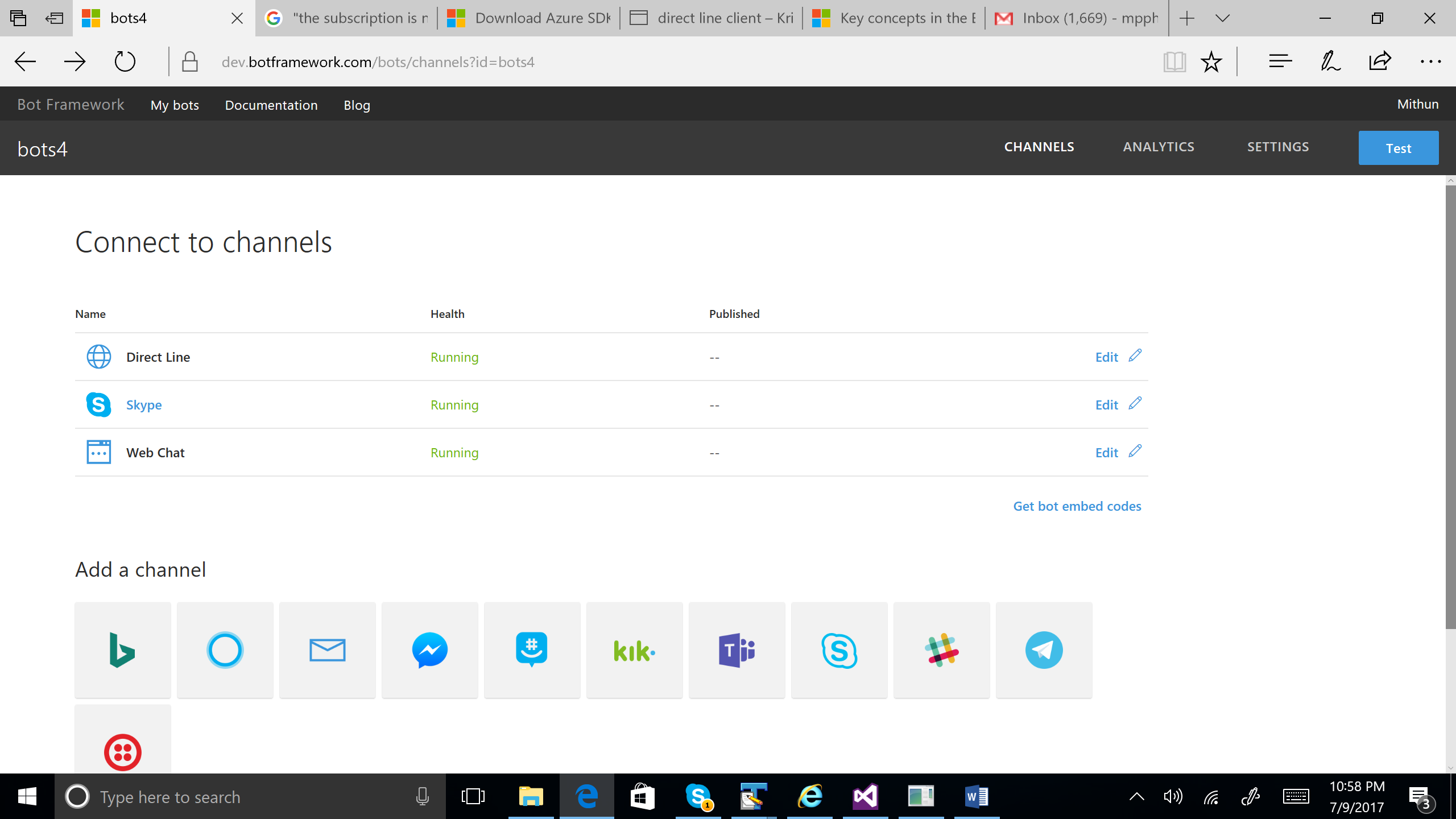
<https://github.com/Microsoft/BotBuilder-Samples/tree/master/CSharp/core-DirectLine> and import the solution in Visual Studio.

In the DirectLineBot solution, you will find two projects – DirectLineBot and DirectLineSampleClient. If you do not have a published app, you can publish DirectLineBot. The previous labs should have prepared you with the steps related to publishing bots.

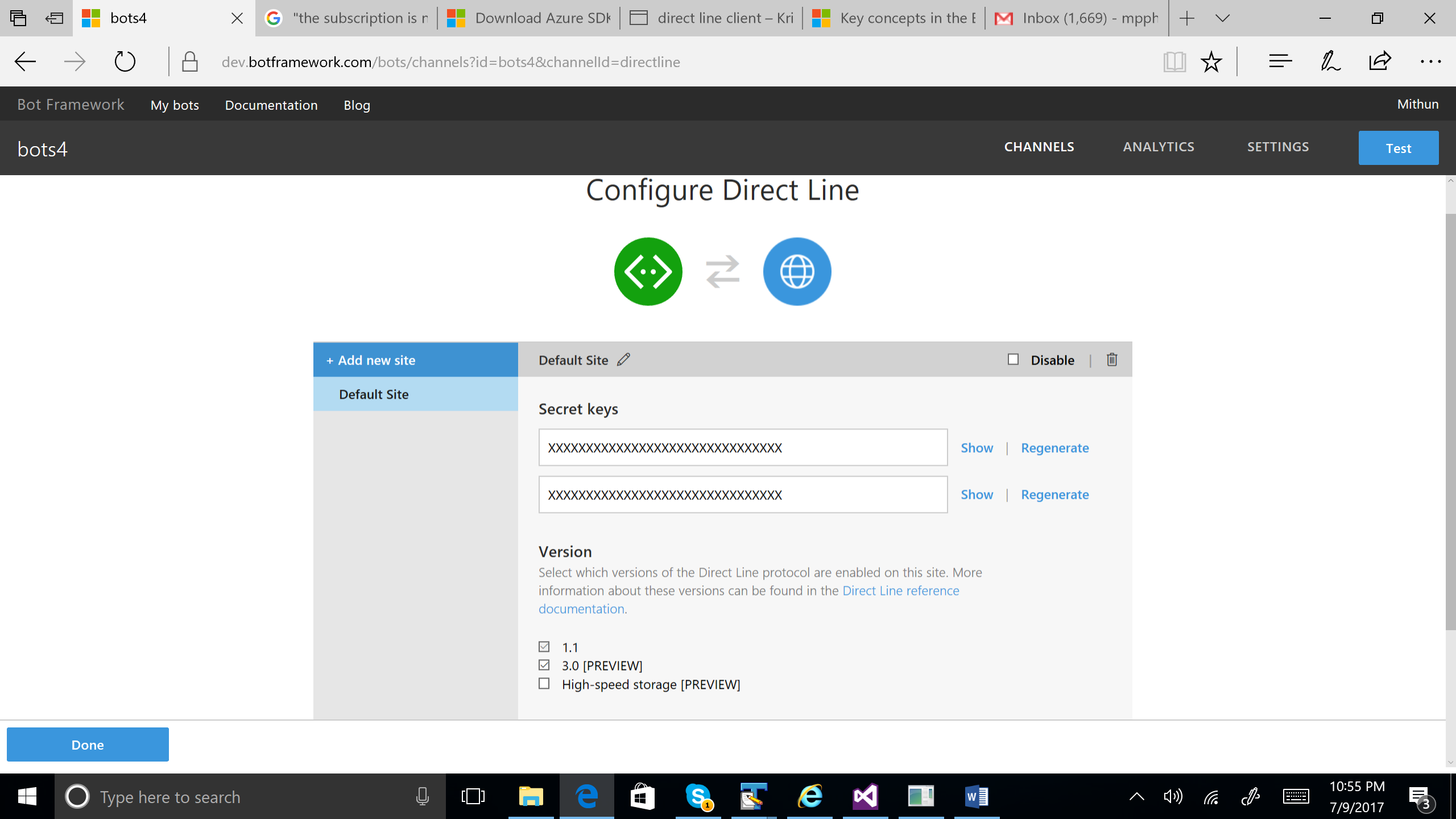
DirectLineSampleClient is the client that will send messages to the bot.

Authentication

Direct Line API requests can be authenticated either by using a secret that you obtain from the Direct Line channel configuration page in the [Bot Framework Portal](https://dev.botframework.com/). Go to the [Bot Framework Portal](https://dev.botframework.com/) and find your bot. Add Direct Line via *Connect to channels* for your bot.



You can obtain a Secret Key from the Direct Line channel after adding as shown below:

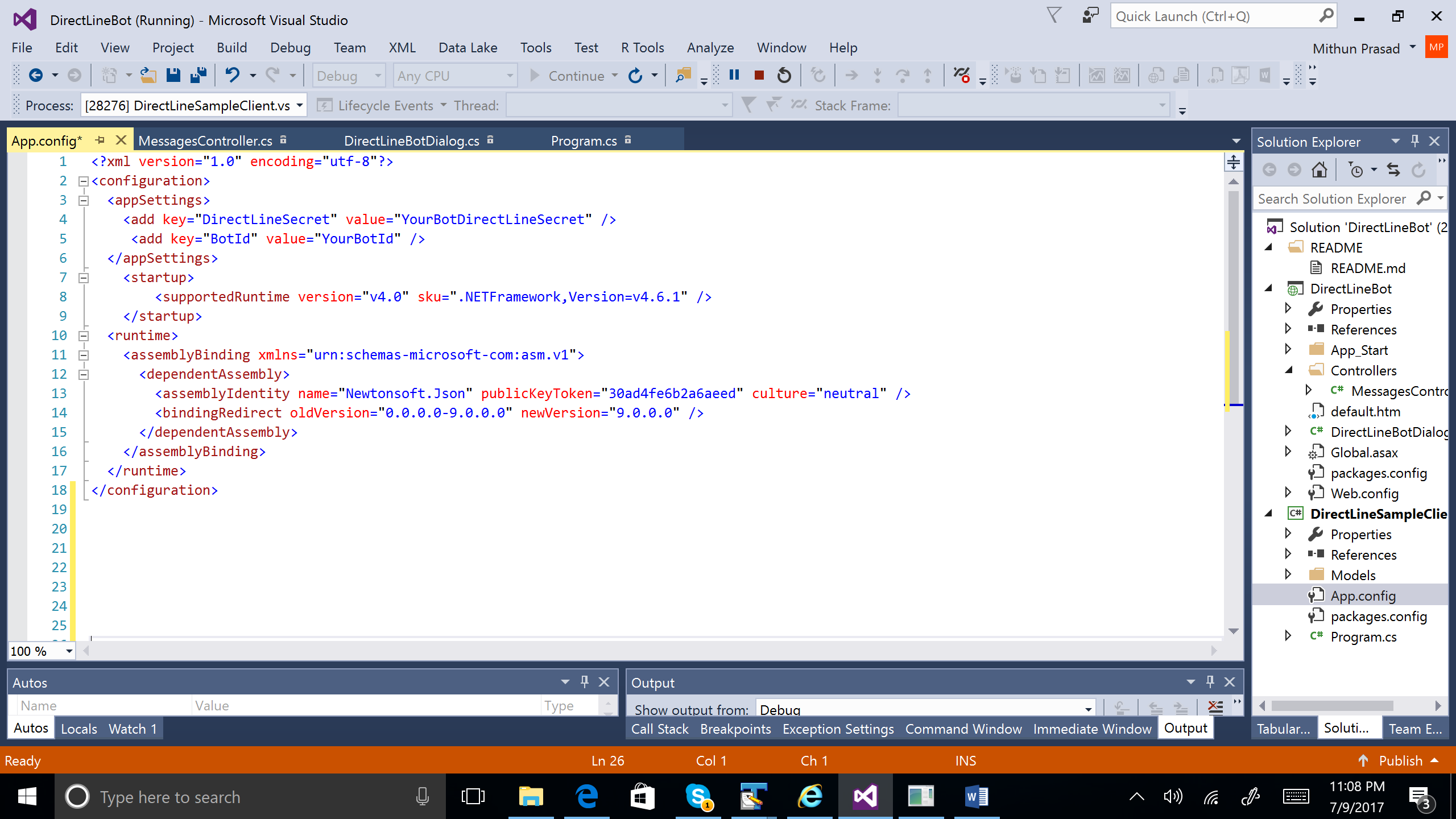


App Config

The Secret key obtained from *Configure Direct Line* in the [Bot Framework Portal](https://dev.botframework.com/) is then added to the Configuration settings in App.config file as shown below. In addition, for the published bot, capture the bot id and enter in the appSettings part of App.config from DirectLineSampleClient project. The relevant lines of App.config to enter in the App.config are listed as follows:

<add key="DirectLineSecret" value="YourBotDirectLineSecret" />

<add key="BotId" value="YourBotId" />

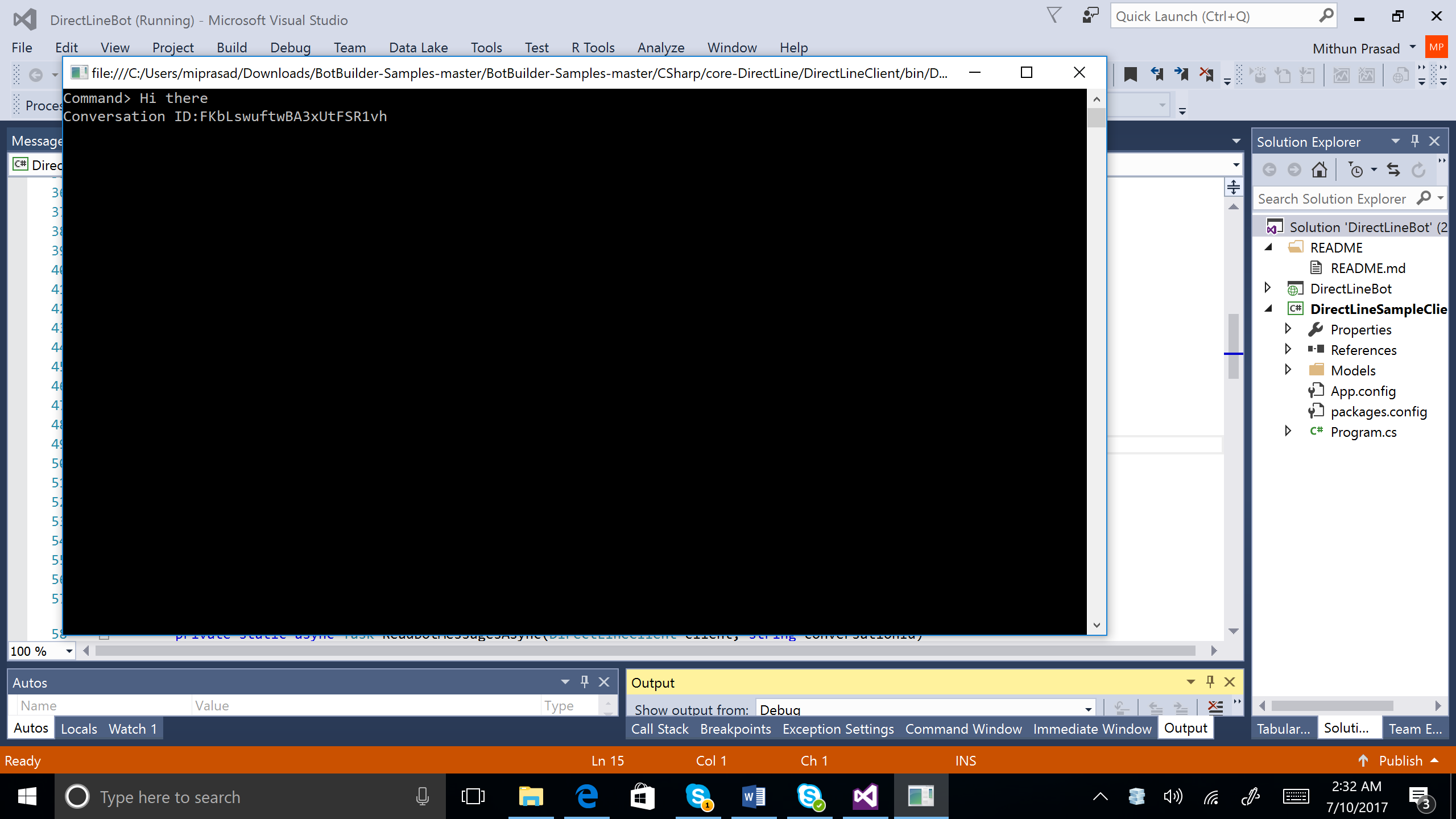


Sending and Receiving Messages

Using Direct Line API, a client can send messages to your bot by issuing HTTP Post requests. A client can receive messages from your bot either via WebSocket stream or by issuing HTTP GET requests. In this lab, we will explore HTTP Get option to receive messages.

1. Run project DirectLineSampleClient after making the Config changes.
2. Submit a message via console and obtain the conversation id using the below line:

Console.WriteLine("Conversation ID:" + conversation.ConversationId);



1. Once you have the conversation id, you can retrieve user and bot messages using HTTP Get. To retrieve messages for a specific conversation, you can issue a GET request to the

/api/conversations/{conversationId}/activities endpoint, optionally specifying the watermark parameter to indicate the most recent message seen by the client.

1. Use any Rest Client to receive messages via HTTP Get. In this lab, <https://advancedrestclient.com/> is used for receiving messages from the bot. The below images indicate the conversations obtained from *Advanced Rest client*. Note the conversation “Hi there” and the corresponding bot response that is echoed back.

