@HJaimesDEv | Humberto@humbertojaimes.net

Microsoft Bot Framework + Xamarin

Humberto Jaimes

2017

Table of Contents

[Preparando el ambiente. 3](#_Toc475321157)

[Crear el bot 4](#_Toc475321158)

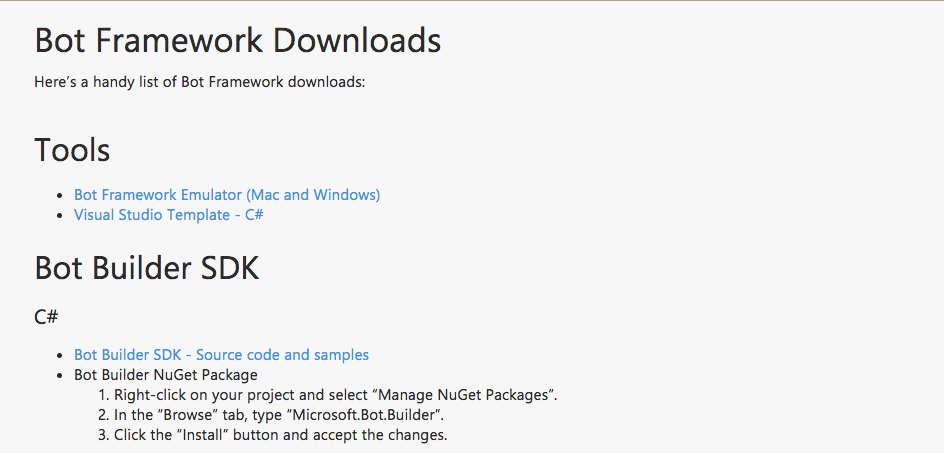
[Publicar el bot 9](#_Toc475321159)

[Registrando el bot 11](#_Toc475321160)

[Consumiendo el bot desde Xamarin 16](#_Toc475321161)

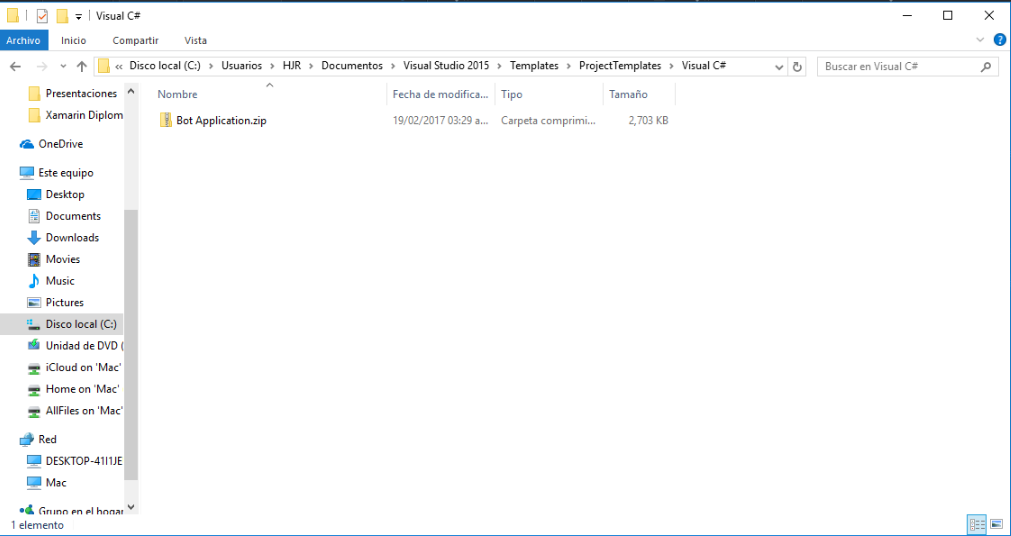
# Preparando el ambiente.

En el sitio <https://docs.botframework.com/en-us/downloads/> se debe descargar la plantilla para Visual Studio y el Bot Framework Emulator.



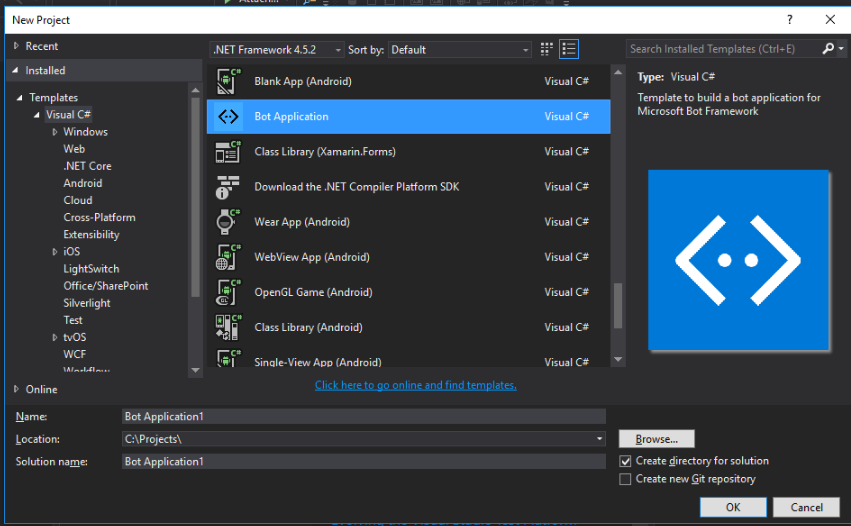
Para instalar la plantilla en Visual Studio hay que pegar el .zip descargado en la siguiente ruta

C:\Users\TU USUARIO\Documents\Visual Studio 2015\Templates\ProjectTemplates\Visual C#

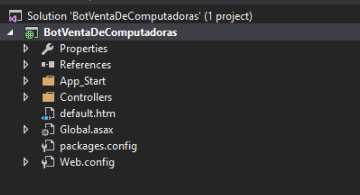


# Crear el bot

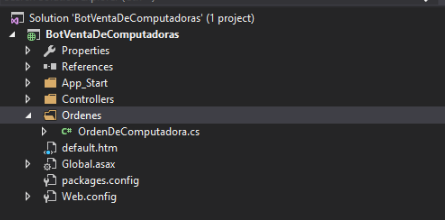
Después de poner el archivo en el lugar correspondiente Ya podremos crear proyectos del Bot Framework desde Visual Studio, la platilla se encuentra dentro de la sección Visual C#.



Para crear nuestro primer bot crea un proyecto nuevo con el nombre de BotVentaDeComputadoras.



Dentro del proyecto generado crea una carpeta llamada “Ordenes” y dentro una clase llamada “OrdenDeComputadora”

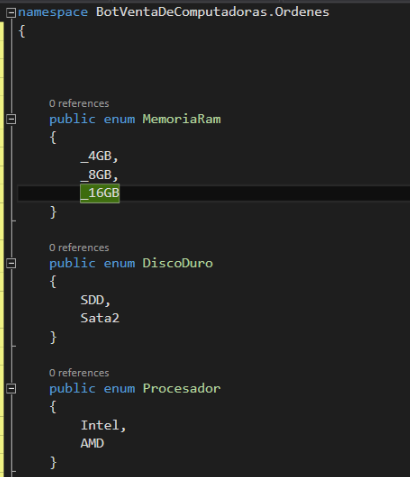


Cuando utilizamos el FormFlow tenemos la facilidad de crear una conversación guiada a partir de una clase. La idea es muy simple, esta clase va a ser la representación de una serie de campos que el usuario debe llenar, como si se tratara de un formulario.

El FormFlow puede trabajar con los siguientes tipos de campo

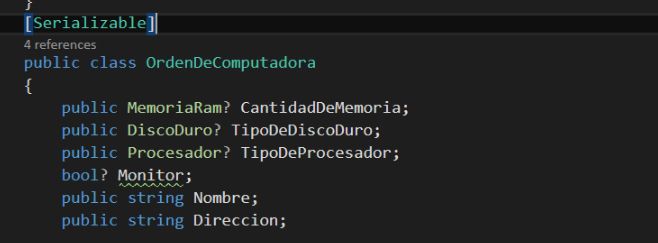
* sbyte, byte, short, ushort, int, uint, long, ulong
* float, double
* String
* DateTime
* Enum
* List of enum

Dentro del namespace crea algunas enumeraciones, por ejemplo, las siguientes:



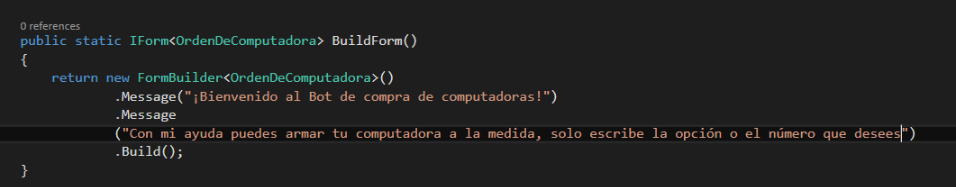
Estas enumeraciones servirán para definir las opciones validas que puede elegir el usuario.

Ahora crea los campos necesarios para la conversación dentro de la clase, si se usa el estándar de nomenclatura CamelCase, FormFlow puede detectar las palabras a la hora de crear la conversación.

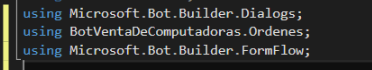


Dentro de la clase crea el método necesario para generar el formulario. Recuerda que un bot debe presentarse y explicar cuál es su función, para esto se utiliza el método Message que nos permite enviar un mensaje antes de construir nuestro formulario con los campos que definimos.

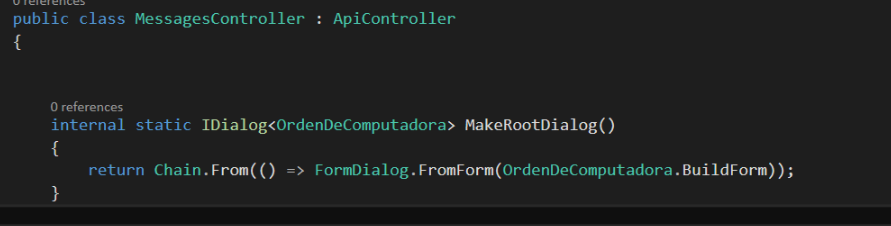
Las siguientes líneas requieren del importar el namespace Microsoft.Bot.Builder.FormFlow, debes agregar el using correspondiente a la clase.



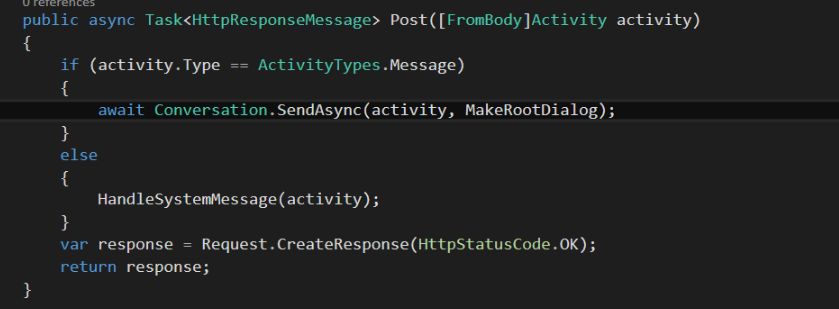
Para poder utilizar el bot debes hacer unos cambios al controlador generado por la plantilla, dentro del archivo MessageController que se encuentra dentro de la carpeta Controllers agrega los siguientes namespaces



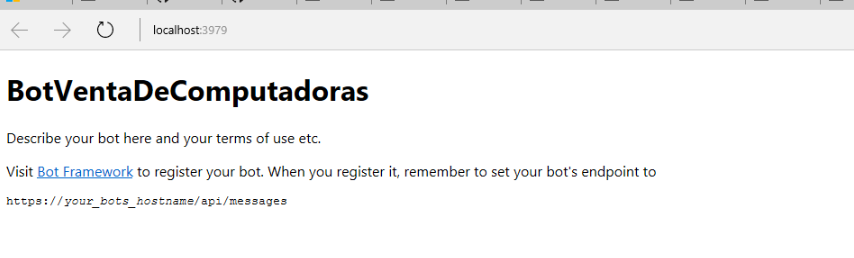
Ahora dentro de la clase MessageController agrega las siguientes líneas encargadas de generar los diálogos a partir del formulario creado con FormFlow



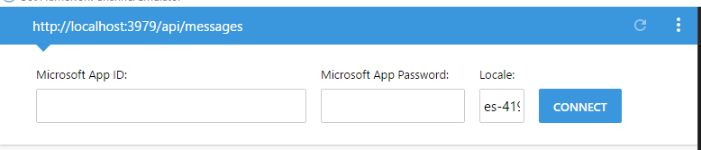
Para finalizar modifica la condición dentro método Post en el mismo controlador, al modificarlo debe quedar así

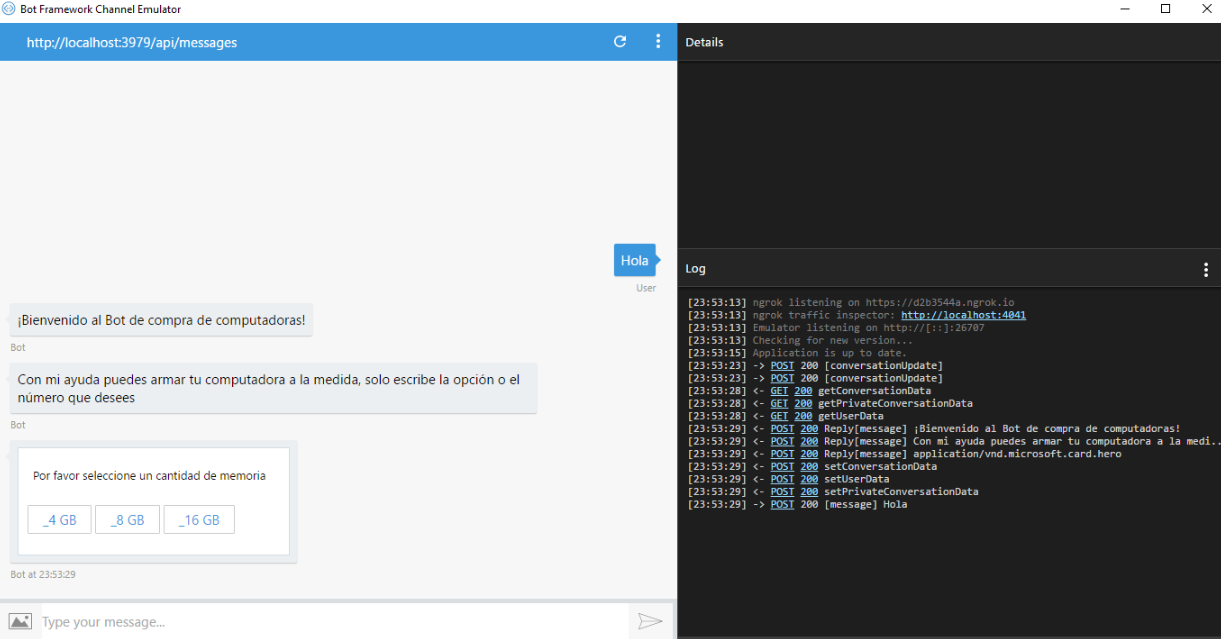


Para probar nuestro bot debemos ejecutarlo, si todo es correcto se abrirá el explorador web mostrando lo siguiente



Ahora conectando el Bot Emulator con el puerto indicado podemos comenzar a interactuar con el bot.

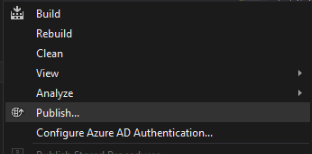




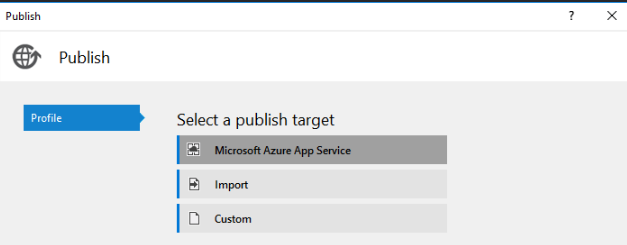
# Publicar el bot

Al concluir las pruebas, podemos publicar nuestro bot en Microsoft Azure, el proceso es el siguiente:

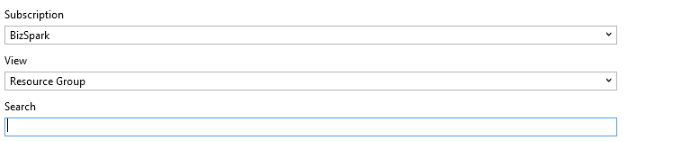
Dando clic derecho sobre el proyecto selecciona “Publish”



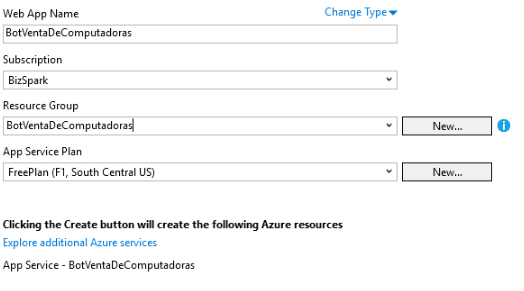
En la nueva ventana selecciona “Microsoft Azure App Service”



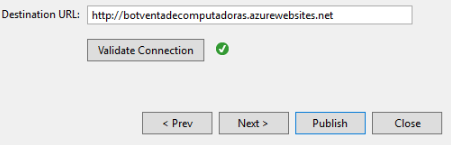
Llena los datos que se solicitan de acuerdo a tu cuenta de Microsoft Azure



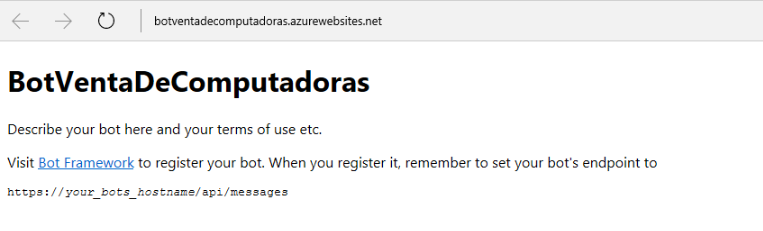
Genera un nuevo Resource Group y captura los datos para tu nuevo bot, puedes seleccionar el plan gratuito.



Finalmente verifica la conexión y selecciona “Publish”

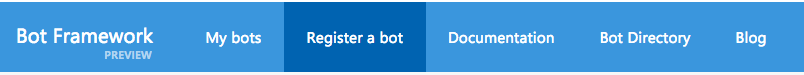


Al finalizar la publicación en el navegador vamos a ver el sitio con nuestro servicio

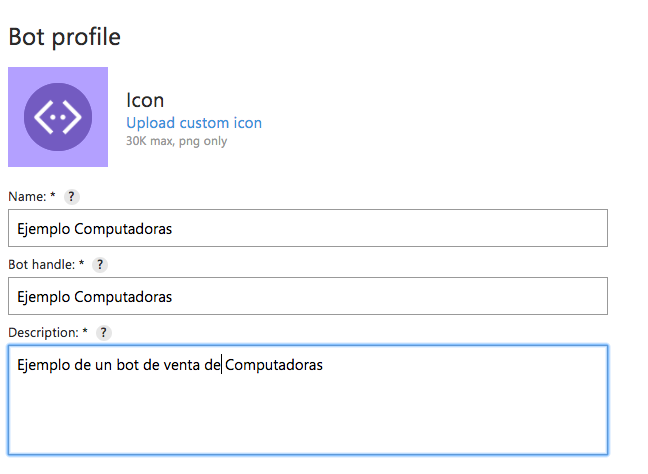


# Registrando el bot

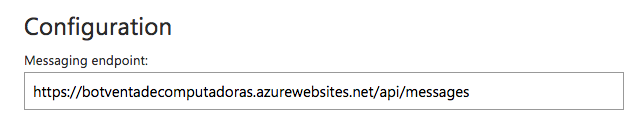
Para poder usar nuestro bot desde diferentes canales (Facebook, Twitter, etc.) debemos registrarlo, esto se hace en <https://bots.botframework.com> donde debes iniciar sesión con una cuenta Microsoft. Después de iniciar sesión elige “Register a bot”



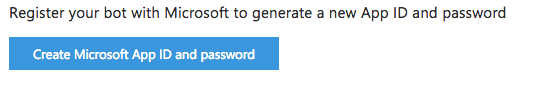
Llena la descripción de tu bot



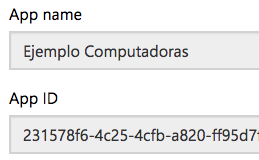
Luego completa y genera los datos de conexión hacia tu bot, primero pega la dirección https donde esta hospedado tu bot



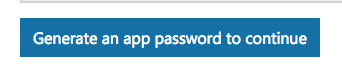
Genera el AppID y el password presionando el botón



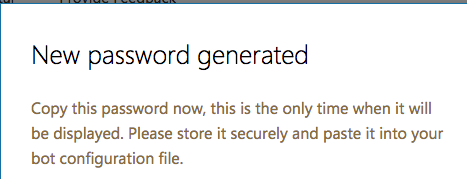
Copia el App ID generado y guárdalo temporalmente



Ahora presiona el botón para generar el password



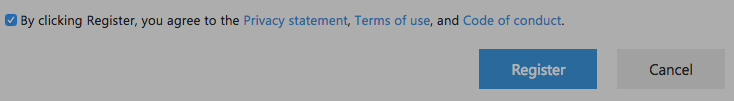
Al igual que con el App ID guarda el password generado y presiona el botón OK



Ahora vuelve al registro del bot

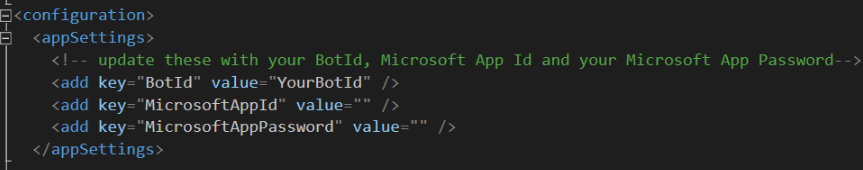


Lee y acepta las condiciones y crea tu nuevo bot



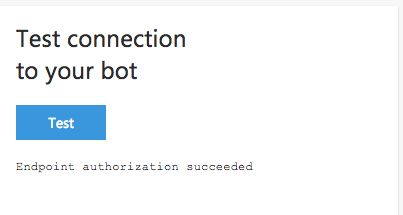


Al terminar de registrar tu bot debes agregar el App ID y el password al Web.Config de tu proyecto en Visual Studio, esto se hace en la siguiente sección

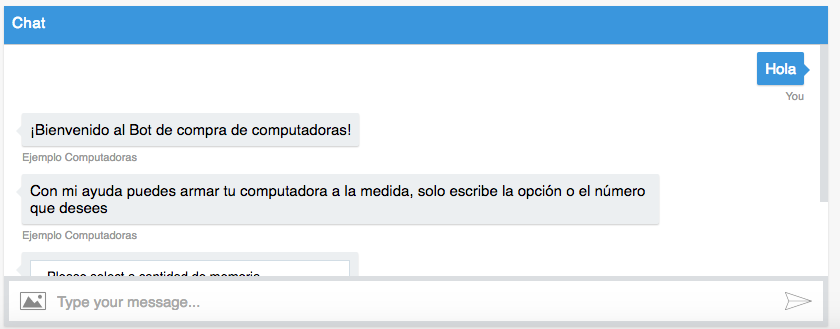


Después de agregar los datos, vuelve a publicar tu bot en Azure.

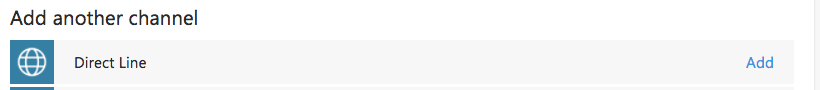
Regresa al portal de registro de bots y prueba la conexión



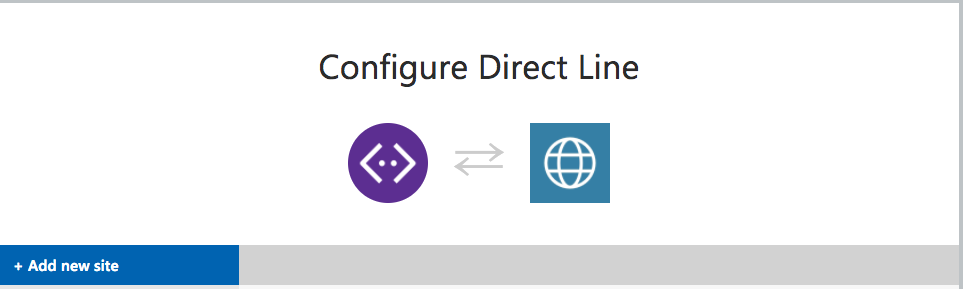
En caso de éxito podemos volver a hacer pruebas desde el portal



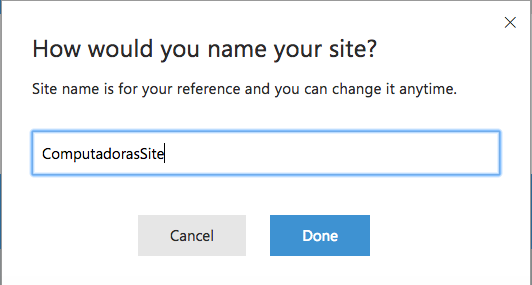
Para poder consumir nuestro bot desde Xamarin debes activar el canal de DirectLine, el cual permite la comunicación con el bot mediante un Api Rest



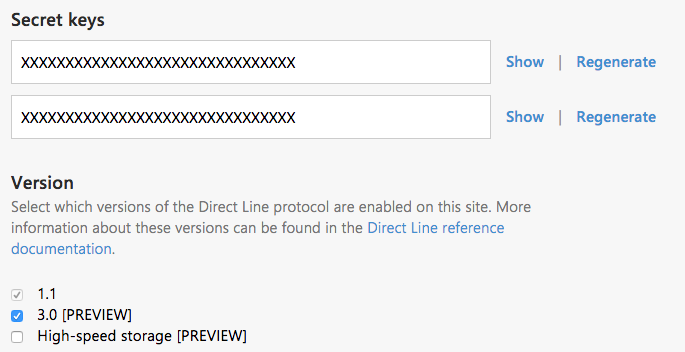
Selecciona “Add” y en la siguiente pantalla elige “Add new site”



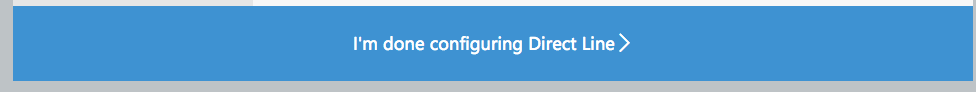
Define el nombre que quieras y presiona “Done”



Las llaves generadas son las que usaras para la comunicación con el bot desde el cliente Rest



Finaliza presionando el siguiente botón

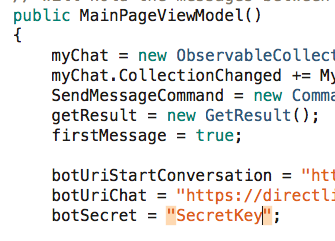


# Consumiendo el bot desde Xamarin

Si quieres probar tu bot en una app Xamarin puedes usar el siguiente ejemplo de proyecto

(<https://github.com/humbertojaimes/XamarinFormsBot>)este proyecto está basado en el proyecto creado por Jorge Cupi (<https://github.com/JorgeCupi/XamarinFormsBot>) y tiene una implementación para soportar las respuestas enviadas por el bot creado a lo largo de este ejercicio.

Solo debes poner tu Secret Key en el constructor del MainPageViewModel



Así se ve el ejemplo funcionando

