EBook Gratis

APRENDIZAJE botframework

Free unaffiliated eBook created from
Stack Overflow contributors.

#botframework

Tabla de contenido

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: botframework

It is an unofficial and free botframework ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official botframework.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

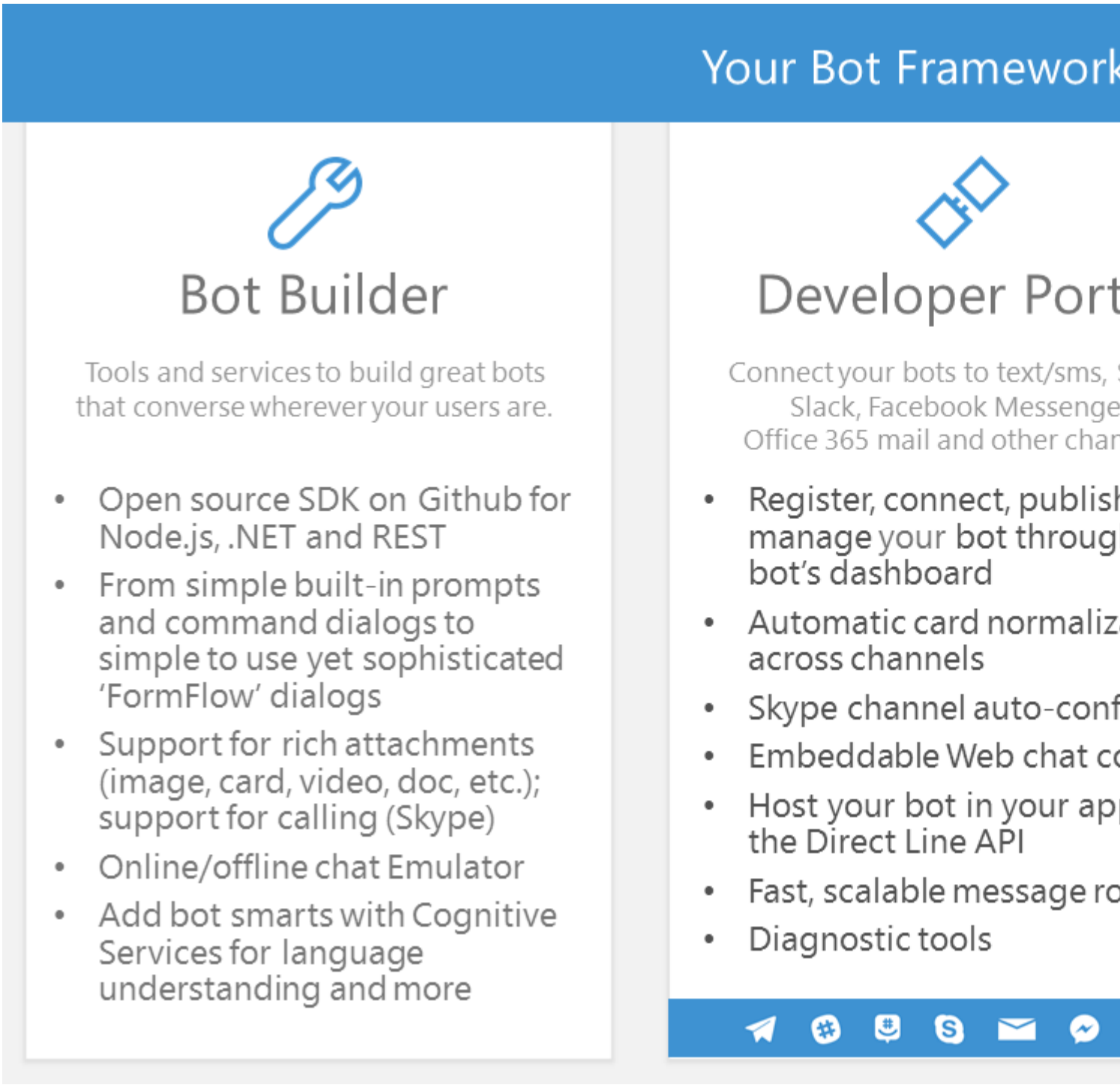
Capítulo 1: Empezando con botframework

Observaciones

Microsoft Bot Framework es una oferta integral para crear e implementar bots de alta calidad para que sus usuarios disfruten de sus experiencias de conversación favoritas. Los desarrolladores que escriben bots se enfrentan a los mismos problemas: los bots requieren una E / S básica; deben tener habilidades de lenguaje y diálogo; deben ser ejecutantes, receptivos y escalables; y deben conectarse a los usuarios, idealmente en cualquier experiencia de conversación e idioma que el usuario elija. Bot Framework le ofrece exactamente lo que necesita para crear, conectar, administrar y publicar bots inteligentes que interactúan de forma natural dondequiera que hablen sus usuarios, desde SMS / texto a Skype, Slack, Facebook Messenger, Kik, Office 365 y otros servicios populares.

Los bots (o agentes de conversación) se están convirtiendo rápidamente en una parte integral de la experiencia digital: son una forma vital para que los usuarios interactúen con un servicio o aplicación, al igual que un sitio web o una experiencia móvil. Los desarrolladores que escriben bots se enfrentan a los mismos problemas: los bots requieren una E / S básica; deben tener habilidades de lenguaje y diálogo; y deben conectarse a los usuarios, preferiblemente en cualquier experiencia de conversación e idioma que el usuario elija. Bot Framework proporciona herramientas para resolver fácilmente estos problemas y más para los desarrolladores, por ejemplo, traducción automática a más de 30 idiomas, administración de estado de conversación y usuario, herramientas de depuración, un control de chat web incrustado y una forma para que los usuarios descubran, prueben y agreguen A los bots les encanta las experiencias de conversación.

Bot Framework consta de una serie de componentes que incluyen el SDK de Bu Builder, el Portal de desarrolladores y el Directorio de Bot.



Versiones

Últimos lanzamientos de Bot Builder

Idioma	Versión	Fecha de lanzamiento
Node.js	3.7.0	2017-02-23
DO#	3.5.5	2017-03-07

Las versiones anteriores se pueden encontrar aquí .

Examples

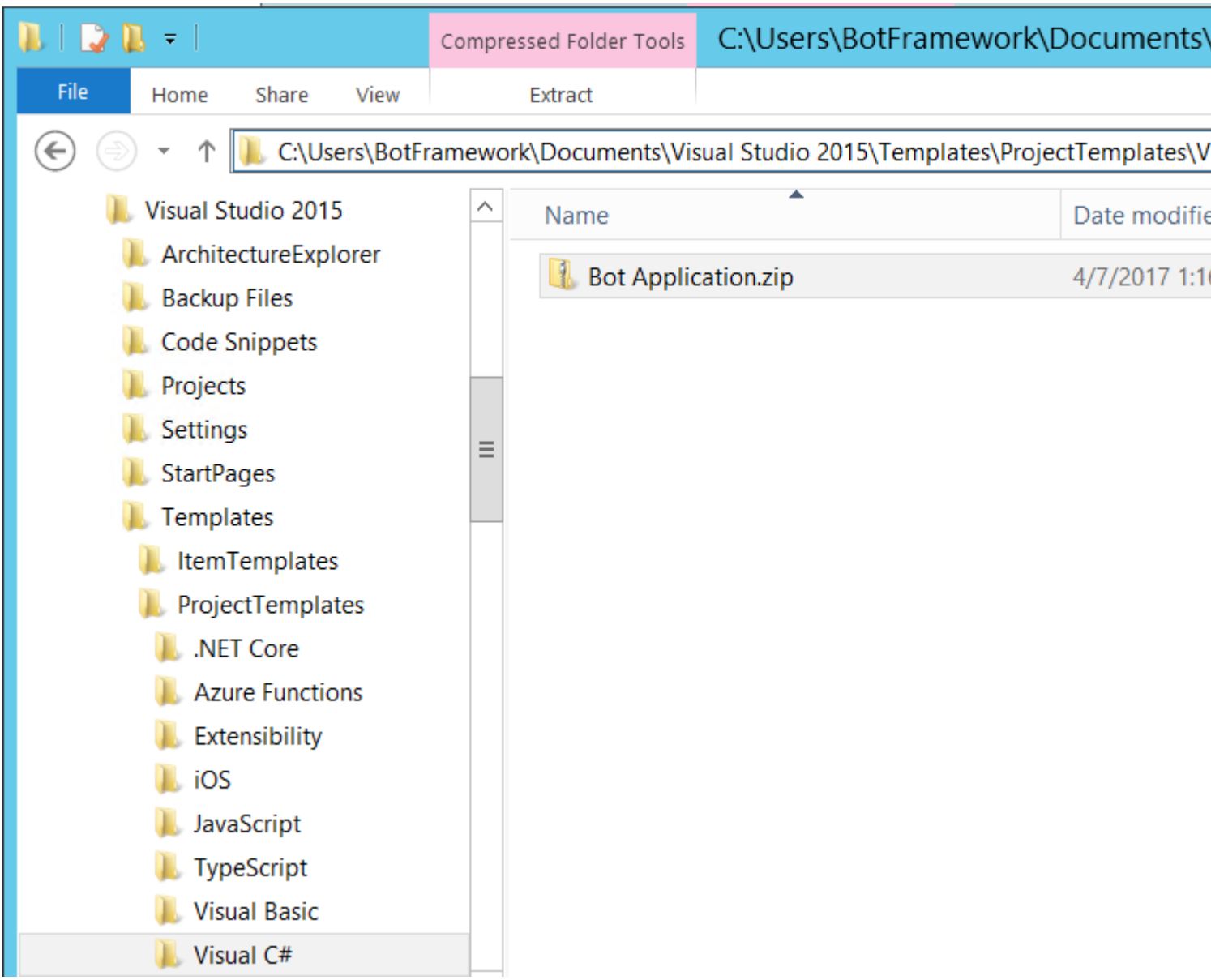
Instalación o configuración

DO#

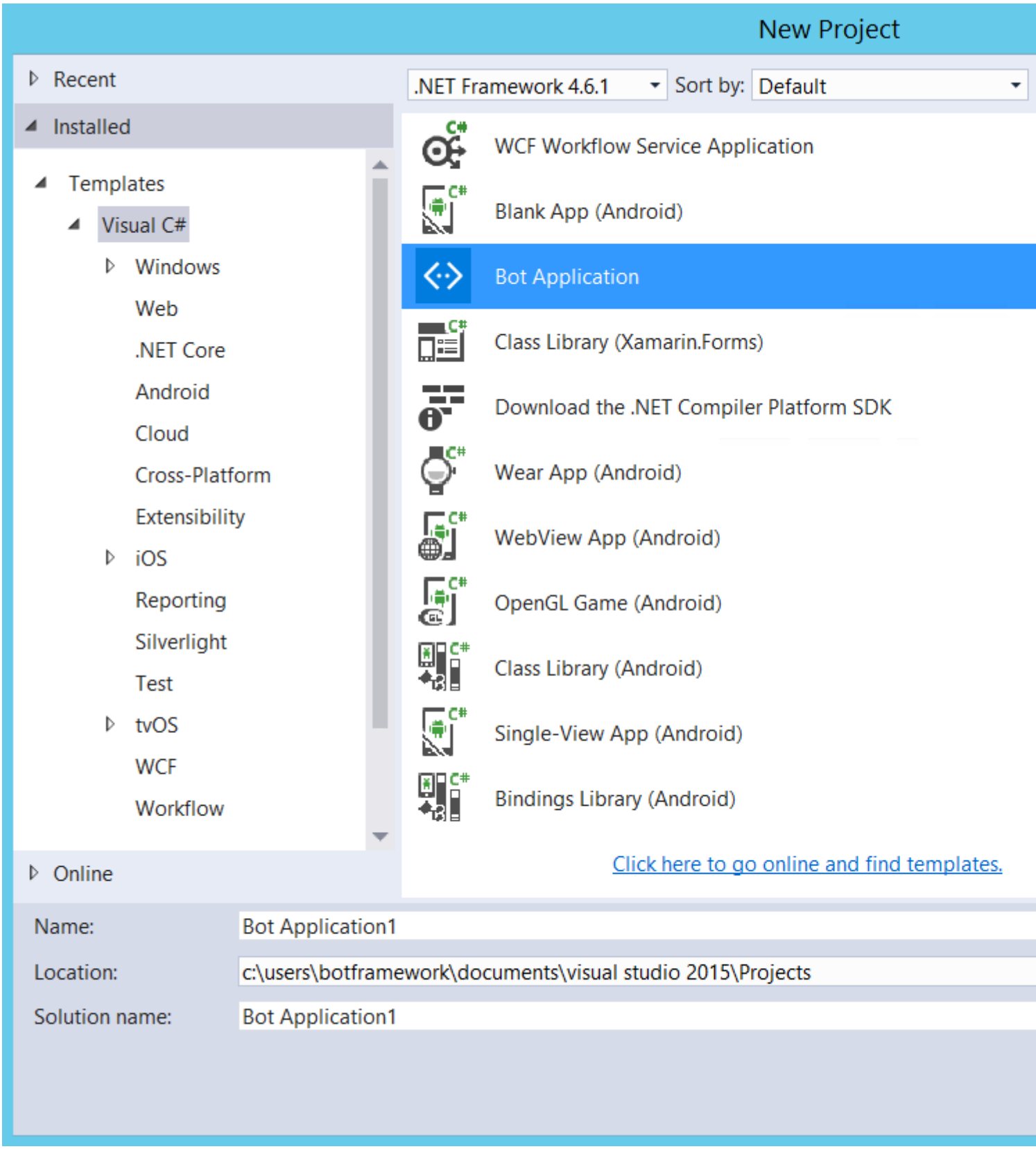
1. **Visual Studio 2015** (última actualización): puede descargar gratis la versión de la comunidad aquí: www.VisualStudio.com

2. Importante: **actualice todas las extensiones VS** a sus últimas versiones Herramientas-> Extensiones y actualizaciones-> Actualizaciones

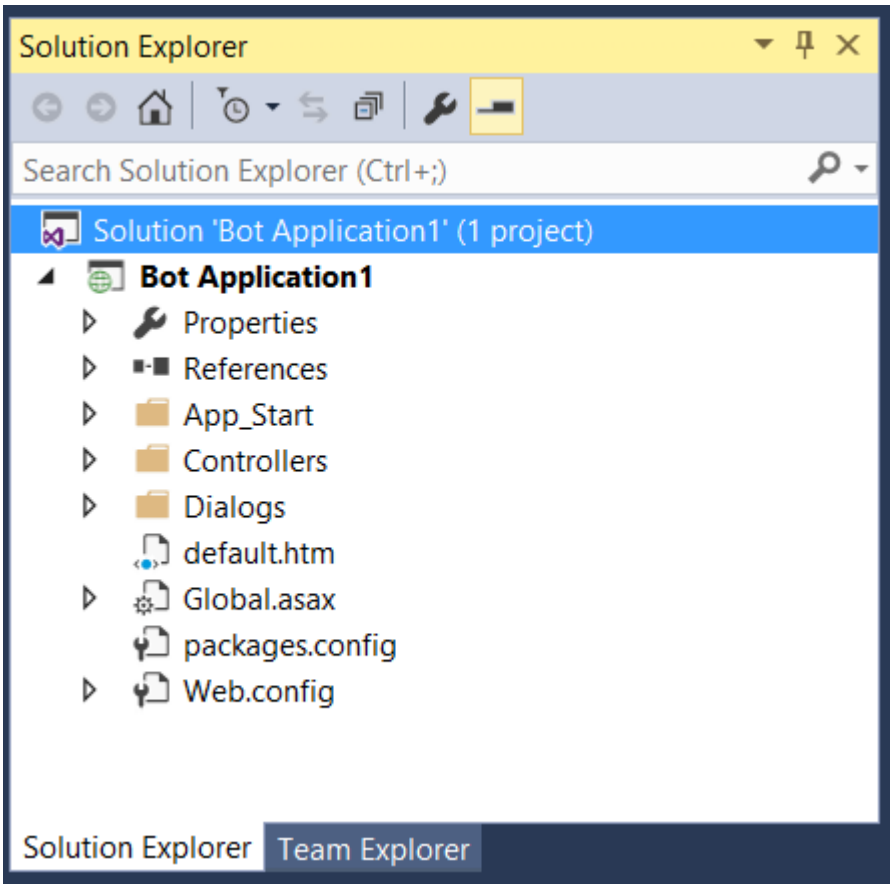
3. Descargue la **plantilla de la aplicación Bot** desde aquí: Descarga de la plantilla Guarde el archivo zip en su directorio de plantillas de Visual Studio 2015 que tradicionalmente se encuentra en "% USERPROFILE% \ Documents \ Visual Studio 2015 \ Templates \ ProjectTemplates \ Visual C #" Nota: deberá reiniciar Estudio visual después de este paso, para poder utilizar la plantilla.



4. Crea un **nuevo proyecto en C** # usando la nueva plantilla de la aplicación Bot



Una vez que tu bot haya terminado de crearse, deberías tener una solución similar a esta:



5. **Ejecute la aplicación** presionando F5 o haciendo clic en el botón verde Ejecutar en la barra de herramientas. Dado que nuestro nuevo bot es en realidad un proyecto WebAPI, se abrirá una ventana del navegador en la página default.htm. El bot ahora se está ejecutando y está expuesto localmente. Tenga en cuenta la url ... será necesario para configurar el bot Framework Emulator en el siguiente paso.

Node.js

1. Cree un nuevo proyecto `npm init` usando `npm init` .
2. Instale el sdk botbuilder y vuelva a verificar utilizando los siguientes comandos npm:

```
npm install --save botbuilder
npm install --save restify
```

3. Para crear su bot, cree un nuevo archivo llamado index.js, y copie el siguiente código para inicializar el bot.

```
var restify = require('restify');
var builder = require('botbuilder');

// Setup Restify Server
var server = restify.createServer();
server.listen(process.env.port || process.env.PORT || 3978, function () {
   console.log('%s listening to %s', server.name, server.url);
});

// Create chat connector for communicating with the Bot Framework Service
var connector = new builder.ChatConnector({
```

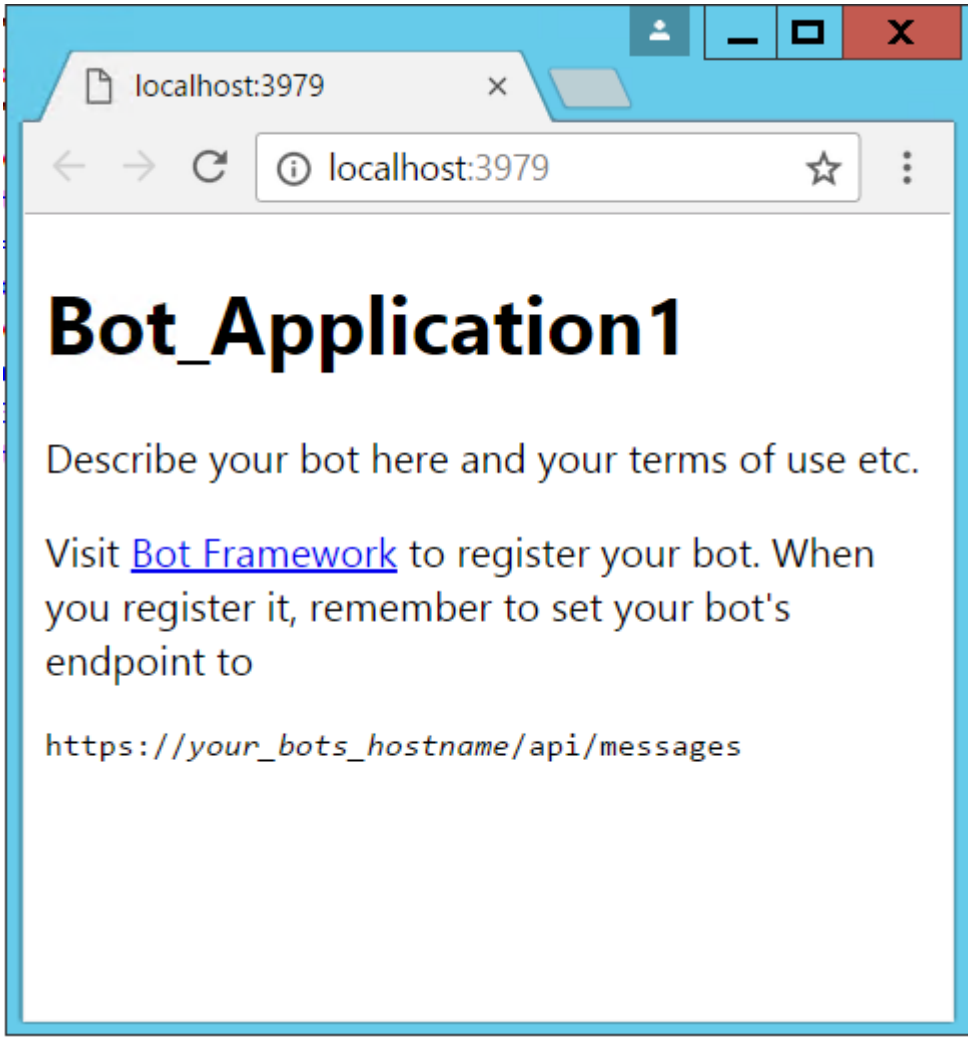
```
    appId: process.env.MICROSOFT_APP_ID,
    appPassword: process.env.MICROSOFT_APP_PASSWORD
});

var bot = new builder.UniversalBot(connector);
```

4. Ahora debería poder ejecutar este archivo utilizando el `node index.js` .

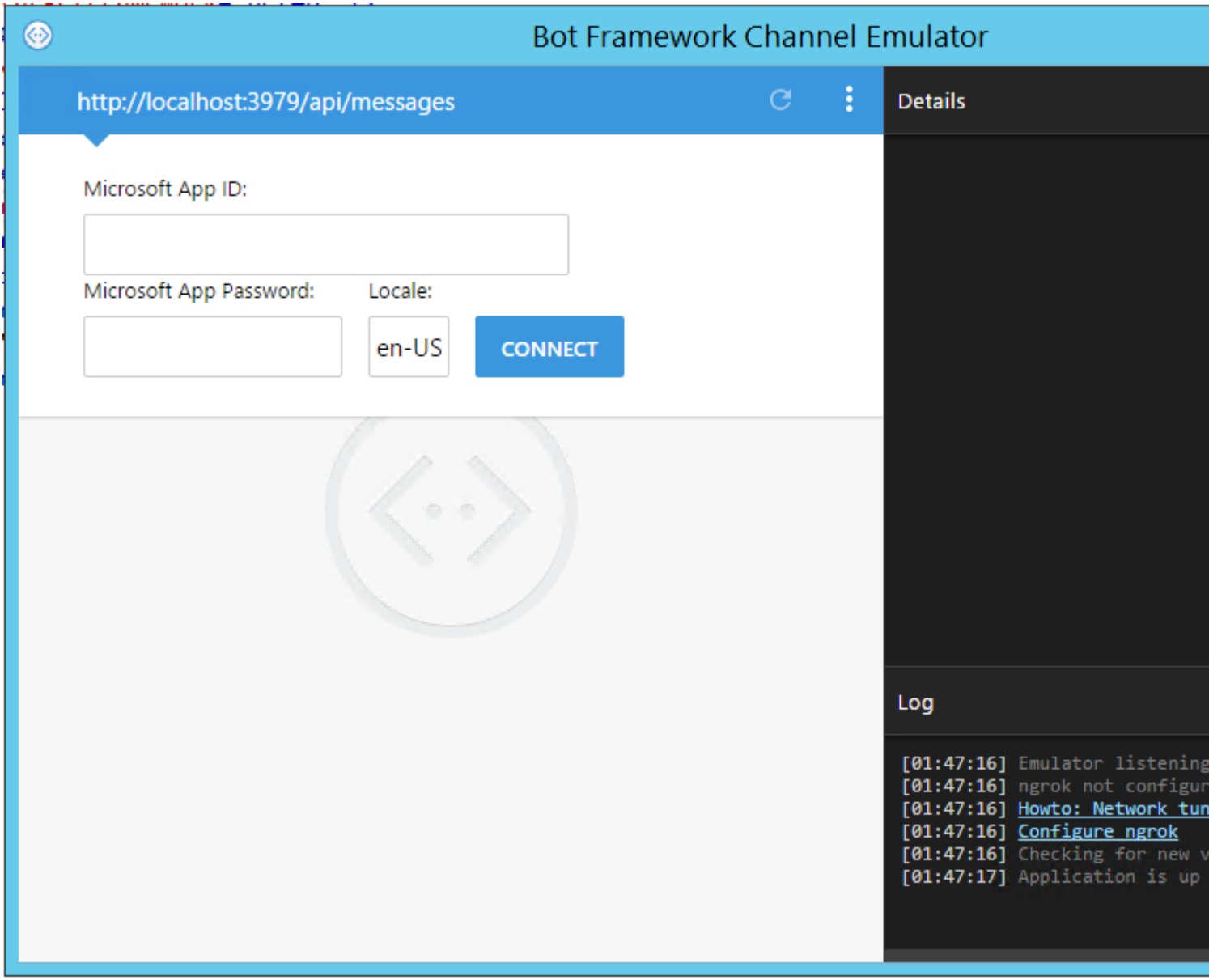
Esta es una configuración básica que será necesaria para todos los bots creados con el framework bot. Puede tratar esto como un proyecto de plantilla en blanco para empezar. Inicializa un servidor de restauración para su bot y crea un conector para conectar máquinas locales con su servidor.

Descargando el emulador para la depuración (tanto para el nodo como para C #)

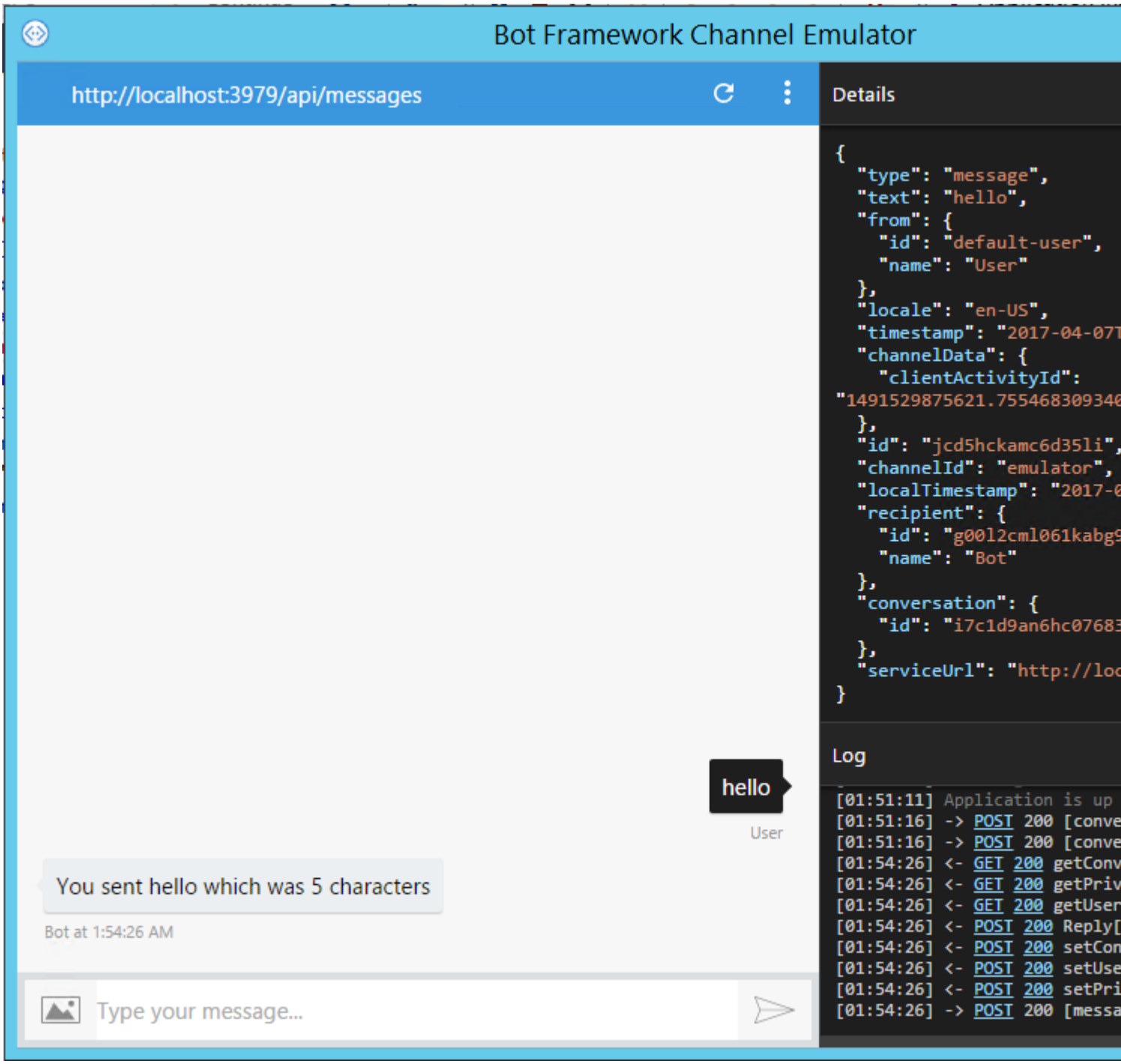


1. Descargue e instale el **simulador de bot Framework** Emulator Descargar

2. Ejecute el emulador e ingrese la url del paso 5 (C #) en el cuadro de texto de la **URL de punto final** . Luego, haga clic en "Conectar".



3. Ahora deberías poder comunicarte con tu bot usando la ventana de chat en el emulador. Verá los detalles de la conversación registrados en la parte inferior derecha, y puede hacer clic en los elementos de línea Publicar y Obtener para ver el json que se ha pasado de un lado a otro.



¡Felicitaciones por crear un Bot usando Microsoft Bot Framework!

Lea Empezando con botframework en línea:
https://riptutorial.com/es/botframework/topic/7509/empezando-con-botframework

Capítulo 2: Adición de procesamiento de lenguaje natural

Introducción

Bot Framework soporta `Recognizers` . Un reconocedor se utiliza para reconocer qué hacer cada vez que un usuario envía un mensaje al bot. Por lo tanto, puede diseñar su bot para reconocer intentos basados en la entrada del usuario. El reconocedor se puede usar con LUIS API para agregar un entendimiento de lenguaje natural para el bot.

Sintaxis

- var reconocizer = new builder.LUISRecognizer ('La URL de su modelo');
- var intents = new builder.IntentDialog ({reconocizer: [reconocizer]});

Examples

Inicializando y agregando LUISRecognizer

Una vez que esté al día con un nuevo proyecto con la plantilla básica que se proporciona en la Introducción, debería poder agregar un Reconocedor de LUIS así:

```
var model = ''    // Your LUIS Endpoint link comes here
var recognizer = new builder.LuisRecognizer(model);
```

Ahora, el `recognizer` es un Reconocedor de LUIS y puede pasar intenciones basadas en su Modelo de LUIS definido. Puede agregar el `recognizer` a sus intenciones por

```
var intents = new builder.IntentDialog({recognizers: [recognizer]});
```

Tu bot ahora es capaz de manejar los intentos de LUIS. Cualquier intentos nombrados sobre Luis se pueden detectar mediante el uso de la `matches` propiedad de `IntentDialog` clase. Así que digamos, una intención llamada `hi` se define en el modelo LUIS, para reconocer la intención en el bot,

```
intents.matches('hi', function(session) {
    session.send("Hey :-)");
});
```

Definiendo un modelo LUIS con intenciones

Crear un modelo LUIS requiere poca o ninguna experiencia en programación. Sin embargo, debe estar familiarizado con 2 términos importantes que se utilizarán ampliamente.

1. **Intenciones** : así se identifican las funciones que deben ejecutarse cuando el usuario escribe algo. Por ejemplo, una intención llamada `Hi` identificará una función que debe ejecutarse cada vez que el usuario envíe "Hola". Los intentos tienen un nombre único en su programa / modelo.
2. **Entidades** : identifican los sustantivos en una declaración. Por ejemplo, "configurar una alarma para 1:00 pm", aquí `1:00 pm` es una entidad que debe ser reconocida por el chat-bot para programar una alarma.

Nota: Las imágenes del sitio web no se proporcionan como la interfaz de mi cambio, pero el concepto central sigue siendo el mismo.

Para crear un nuevo modelo, vaya a LUIS.ai e inicie sesión con su cuenta de Microsoft para acceder a la página de creación de la aplicación. Donde se puede crear un proyecto en blanco.

Definir intenciones:

Los intentos se pueden definir en la pestaña `Intents` . Identifican qué función necesitas realizar cuando el usuario ingresa algo.

Todas las aplicaciones tienen una intención predeterminada `None` , que se activa cuando la entrada del usuario no coincide con ninguna otra intención.

Para definir una intención,

1. Asígnele un nombre único que sea relevante para la función que desea realizar.
2. Una vez que se completa el nombramiento, debe agregar `utterances` a la intención. Las expresiones son lo que desea que el usuario envíe para activar la intención que está definiendo. Intente alimentar la mayor cantidad de `utterances` posibles para que LUIS asocie sus `intents` y `utterances` correctamente.
3. Entrenar a su modelo LUIS, haciendo clic en el `Train` botón de `Train and Test` Tab. Después del entrenamiento, la aplicación se puede probar en el panel de abajo.
4. Finalmente publique su aplicación en la pestaña `Publish App` . Ahora debería obtener una URL de punto final que se debe colocar al definir `LUISRecognizer` en su código bot.

Agregando Entidades al Modelo LUIS

Una entidad es la información que su bot extrae de una expresión particular de acuerdo con una intención.

Por ejemplo, Let `My name is John Doe` pertenece a una intención llamada `introduction` . Para que su robot comprenda y extraiga el nombre `John Doe` de la oración, debe definir una entidad que lo haga. Puede nombrar a la entidad como desee, pero es mejor nombrarla como algo que pertenece a lo que extrae. En nuestro ejemplo, podemos llamar a nuestro `name` entidad.

Las entidades se pueden reutilizar entre diferentes propósitos, para extraer cosas diferentes. Entonces, el mejor principio sería hacer una entidad que extraiga solo el tipo de datos y usarlos en diferentes propósitos. Por lo tanto, en nuestro ejemplo anterior, digamos que `Book a flight on Emirates` pertenece a la intención de la `booking` , luego se puede utilizar la misma entidad, `name` , para extraer los `emirates` nombre del vuelo.

Debe tener en cuenta dos cosas antes de continuar definiendo entidades:

1. Las entidades deben ser únicas por expresión en una intención. Una entidad no puede ser usada dos veces en la misma expresión.
2. LUIS no distingue entre mayúsculas y minúsculas. Esto implica que todo lo extraído y recibido a través de la extracción de la entidad será en minúsculas. Entonces, extraer datos sensibles a mayúsculas y minúsculas a través de entidades es probablemente una mala idea.

Agregando entidades pre-construidas

Las entidades creadas previamente son, como su nombre indica, creadas previamente, es decir, ya están configuradas para extraer un tipo particular de datos a través de la intención a la que se agregan. Un ejemplo puede ser el `number` entidad que extrae números de la intención a la que está asignado. Los números pueden ser numéricos o alfabéticos como `10` o `ten` .

Para obtener una lista completa de todas las entidades predefinidas, puede visitar [Lista de entidades predefinidas] [1].

Para agregar entidades pre-construidas,

1. Ir a la pestaña de `entities` .
2. Haga clic en `Add pre-built entities` y seleccione la entidad que desea agregar al modelo y presione guardar.

Agregar entidades personalizadas **Las entidades** personalizadas son de 4 tipos,

1. **Simple** : una entidad simple extrae un dato particular, el `name` en los ejemplos anteriores es una entidad simple.
2. **Jerárquico** : una entidad principal con entidades secundarias (subtipos) que dependen de la principal.
3. **Compuesto** : Un grupo de 2 o más entidades independientes juntas.
4. **Lista** : una entidad que reconoce palabras solo de una lista dada.

Definiendo Entidades Simples

1. Ve a la pestaña de `entities` .
2. Haga clic en `Add Custom Entities`
3. Nombre su entidad, verifique el tipo de entidad requerida y presione `Save` .

Todos los demás tipos de entidades se pueden agregar de la misma manera simplemente cambiando el `Entity Type` a uno de los tipos anteriores. En los tipos de entidades jerárquicas y compuestas, también deberá proporcionar los nombres de los hijos junto con el nombre de la entidad principal. Definir las entidades de la lista es un poco diferente al resto.

Definir entidades de la lista

Después de seguir los pasos anteriores para crear una `List Entity` colocando el `Entity Type` como lista, se le dirigirá a la página de detalles de la entidad que acaba de definir.

1. Definir un valor canónico. Este es un valor estándar que el bot recibirá cuando el usuario escriba cualquiera de los sinónimos.
2. Definir sinónimos al valor canónico. Se convertirán al valor canónico al ser encontrados por la entidad.

También puede importar listas completas utilizando una matriz de objetos JSON, de la forma:

```
[
    {
        "canonicalForm": "Hey",
        "list": [
            "Howdy",
            "Hi"
         ]
    },
    .
    .
    .
]
```

Asociar una entidad con una intención.

`Pre-built` entidades preconstruidas y de `list` ya tienen un conjunto de valores definidos que pueden extraerse de todas las frases, sin embargo, las expresiones `Simple` , `Hierarchical` y `Composite` deben ser entrenadas para recoger valores.

Esto se puede hacer por

1. Vaya a la pestaña `intents` y elija la intención a la que le gustaría agregar la entidad.
2. Agregue una expresión con un valor ficticio que le gustaría que se extraiga. Oiga, puede agregar `My name is John Doe` como una expresión.
3. Haga clic y arrastre el mouse sobre las palabras que desea que la entidad extraiga. Deberá resaltar a `john doe` en el ejemplo anterior.
4. Se abrirá una lista desplegable con una lista de todas las entidades disponibles en su proyecto. Seleccione el correspondiente como mejor le parezca. `Name` será la entidad seleccionada en el ejemplo anterior.
5. Agregue más expresiones con diferentes valores ficticios cada vez y todas las estructuras posibles que pueda imaginar.
6. Entrena y publica tu modelo LUIS.

Lea Adición de procesamiento de lenguaje natural en línea:
https://riptutorial.com/es/botframework/topic/10004/adicion-de-procesamiento-de-lenguaje-natural

Capítulo 3: Comenzando con el servicio de Azure Bot

Introducción

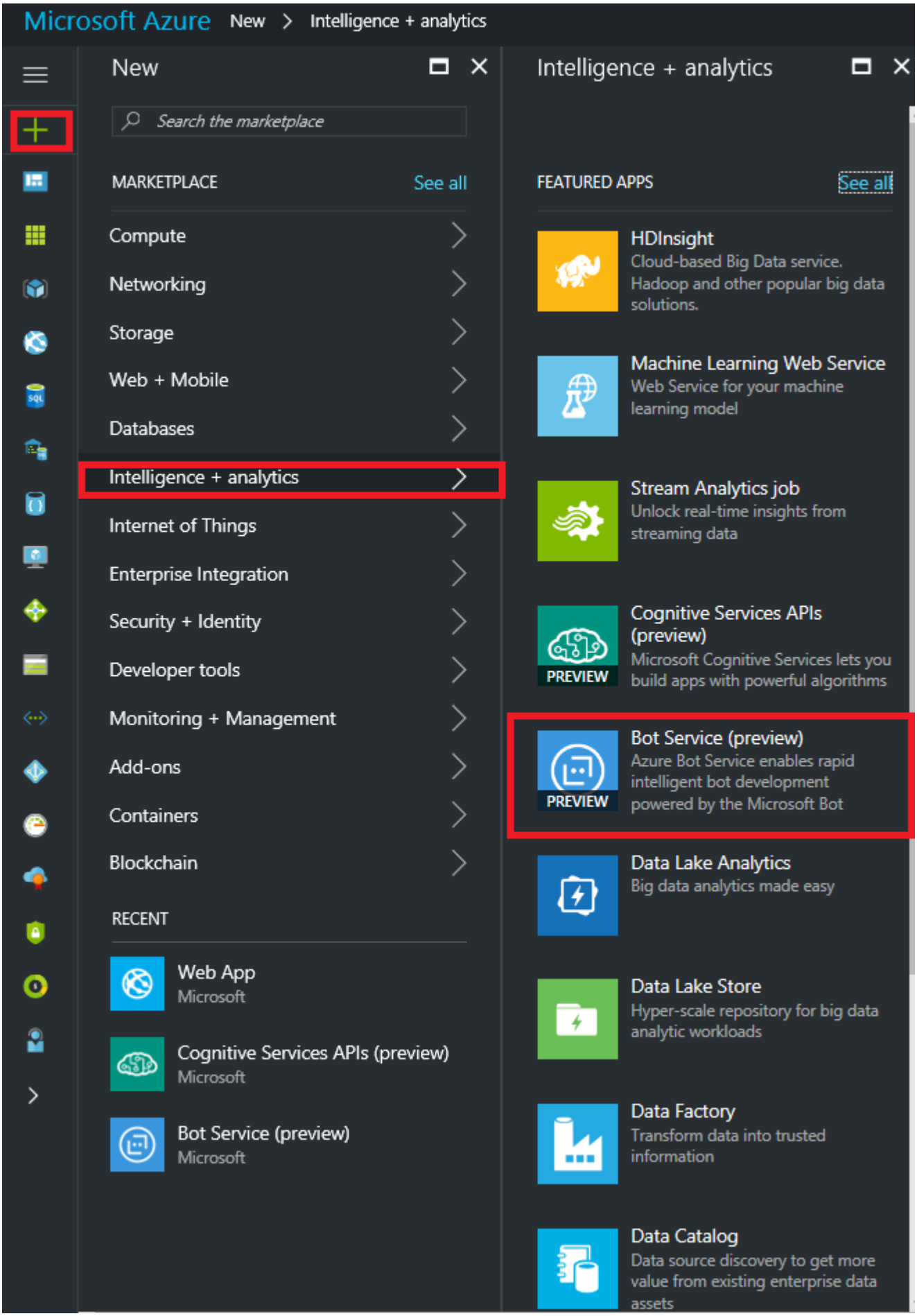
El servicio **Azure Bot** proporciona un entorno integrado que está diseñado específicamente para el desarrollo de bots, lo que le permite construir, conectar, probar, implementar y administrar bots inteligentes, todo desde un solo lugar. Puede escribir su bot en C # o Node.js directamente en el navegador usando el editor de Azure, sin necesidad de una cadena de herramientas. También puede aumentar el valor de sus bots con unas pocas líneas de código conectándose a Cognitive Services para permitir que sus bots puedan ver, escuchar, interpretar e interactuar de manera más humana.

Examples

Comenzando con el servicio de Azure Bot

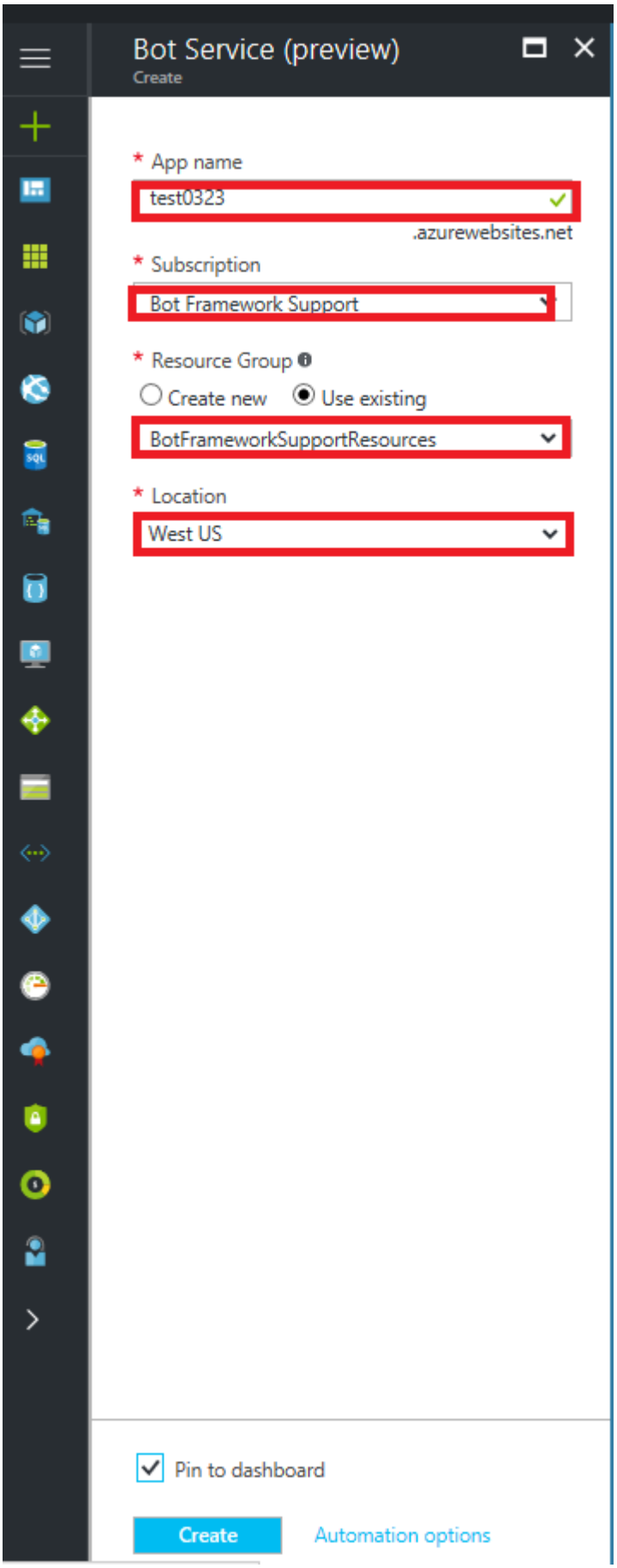
Crea un nuevo bot en Azure siguiendo esta documentación

Inicie sesión en Azure y desde la categoría de Inteligencia y análisis, seleccione Bot Service y proporcione la información requerida.



Ingrese los detalles requeridos para el bot, son idénticos a los detalles requeridos de un Servicio

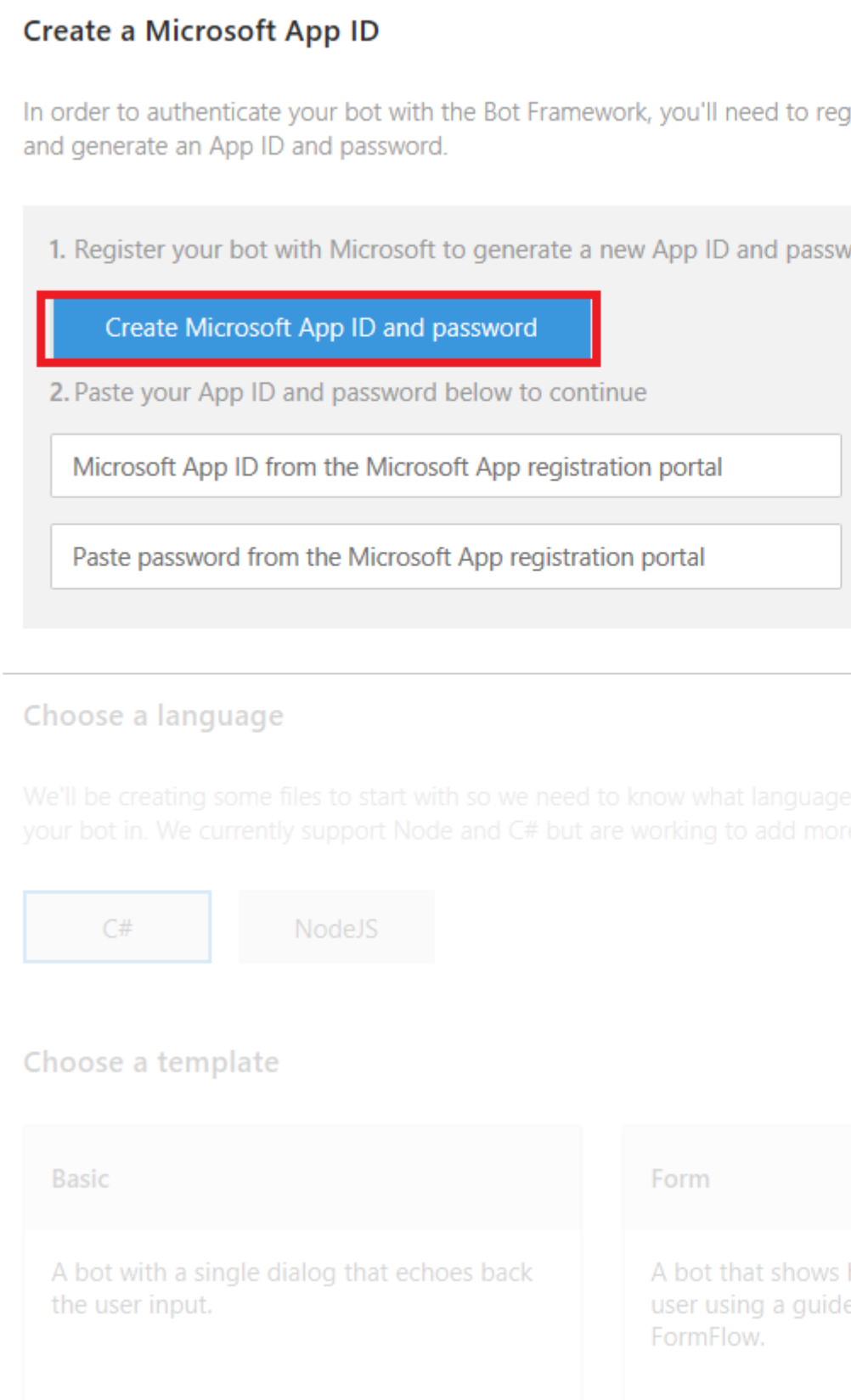
de aplicaciones, por ejemplo, Nombre de la aplicación, Suscripción, Grupo de recursos y Ubicación. Una vez ingresado, haga clic en el botón Crear.



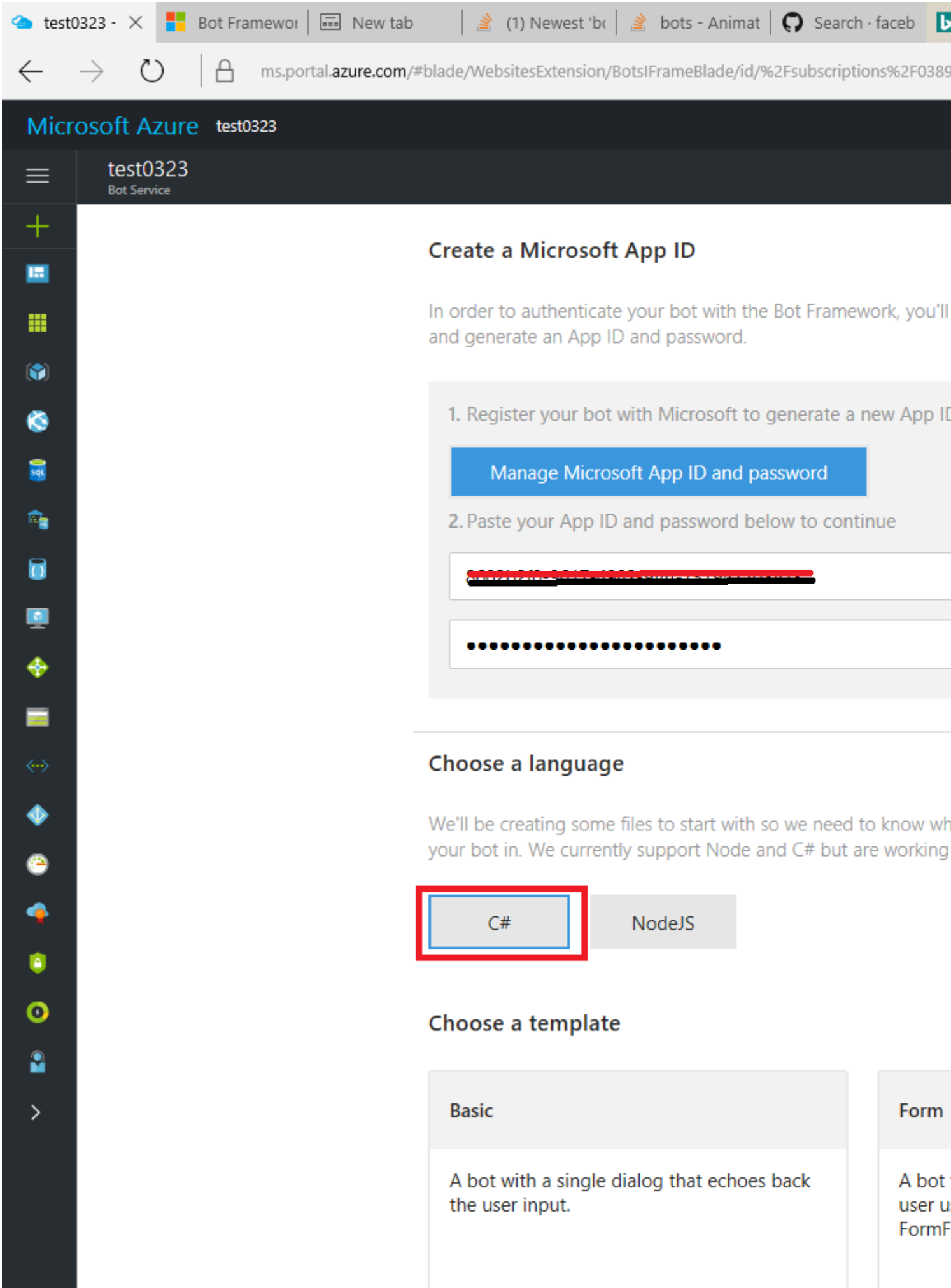
Una vez creado / implementado, navegue hasta el Bot haciendo clic en el enlace desde la página principal, si lo colocó en el tablero o abrió el grupo de recursos y hizo clic en el enlace.

Recuerde que puede haber una pequeña demora antes de que aparezca la pantalla de inicio que indica que el servicio Bot está generando su bot; no haga clic en Crear bot de nuevo.

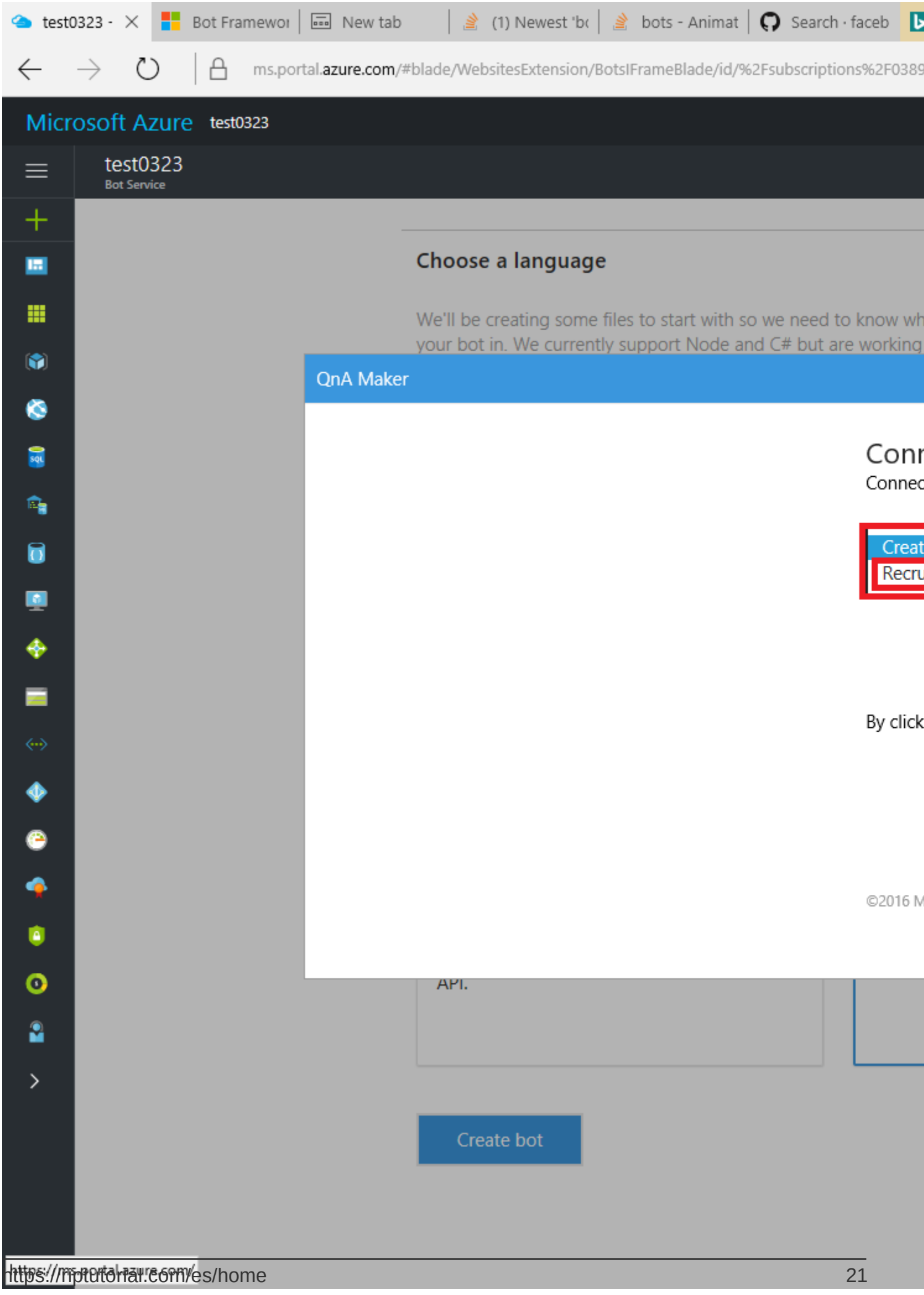
Después de confirmar la implementación, genere y configure el ID de aplicación de Microsoft y la contraseña de la aplicación.



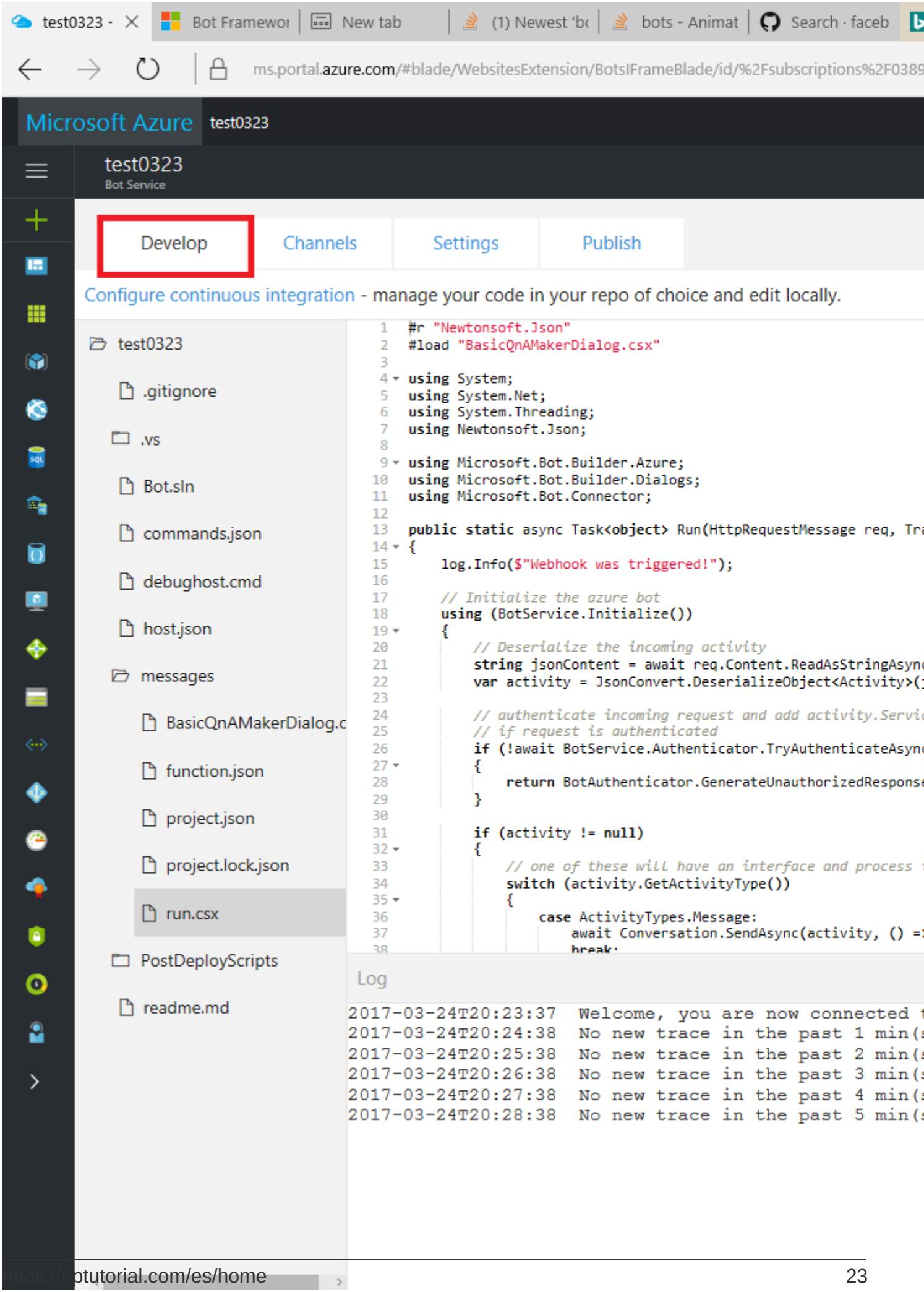
Seleccione el lenguaje de programación de su elección (*seleccioné C #*) y seleccione la **plantilla de preguntas y respuestas** .


test0323 -
Bot Framewo
New tab
(1) Newest 'bo
bots - Animat
Search · faceb
ms.portal.azure.com/#blade/WebsitesExtension/BotsIFrameBlade/id/%2Fsubscriptions%2F0389
Microsoft Azure
test0323
test0323
Bot Service
Create a Microsoft App ID
In order to authenticate your bot with the Bot Framework, you'll
and generate an App ID and password.
1. Register your bot with Microsoft to generate a new App ID
Manage Microsoft App ID and password
2. Paste your App ID and password below to continue
Choose a language
We'll be creating some files to start with so we need to know wh
your bot in. We currently support Node and C# but are working
C#
NodeJS
Choose a template
Basic
A bot with a single dialog that echoes back
the user input.
Form
A bot
user u
FormF

generar una nueva . Como ya había creado una base de conocimientos con mi suscripción, la seleccioné. Esto hizo que mi trabajo fuera mucho más fácil, reduciendo el tiempo requerido para incluir todas las claves en el código del bot de Azure relacionado con la base de conocimientos.


test0323 -
Bot Framewo
New tab
(1) Newest 'bc
bots - Animat
Search · faceb
ms.portal.azure.com/#blade/WebsitesExtension/BotsIFrameBlade/id/%2Fsubscriptions%2F0389
Microsoft Azure
test0323
test0323
Bot Service
Choose a language
We'll be creating some files to start with so we need to know wh
your bot in. We currently support Node and C# but are working
QnA Maker
Conn
Connec
Creat
Recru
By click
©2016 M
API.
Create bot
https://ms.portal.azure.com/

probar el bot funcional en el **control de chat** . El **código predeterminado** se genera al crear el servicio Bot Azure. Puede cambiar la lógica del código en función de sus requisitos.



Cree un nuevo repositorio en el github para configurar la implementación continua con Azure y c

los puntos de interrupción en Visual Studio y ejecutarlos localmente en el emulador y **depurar** siguiendo esta documentación .

Puede hacer un seguimiento de las actualizaciones de compilación y los errores utilizando *Azure Analytics* .

Mirando hacia adelante para actualizar el Bot y pasar al siguiente nivel.

Lea Comenzando con el servicio de Azure Bot en línea:
https://riptutorial.com/es/botframework/topic/9557/comenzando-con-el-servicio-de-azure-bot

Capítulo 4: Comenzando con los servicios de QnA

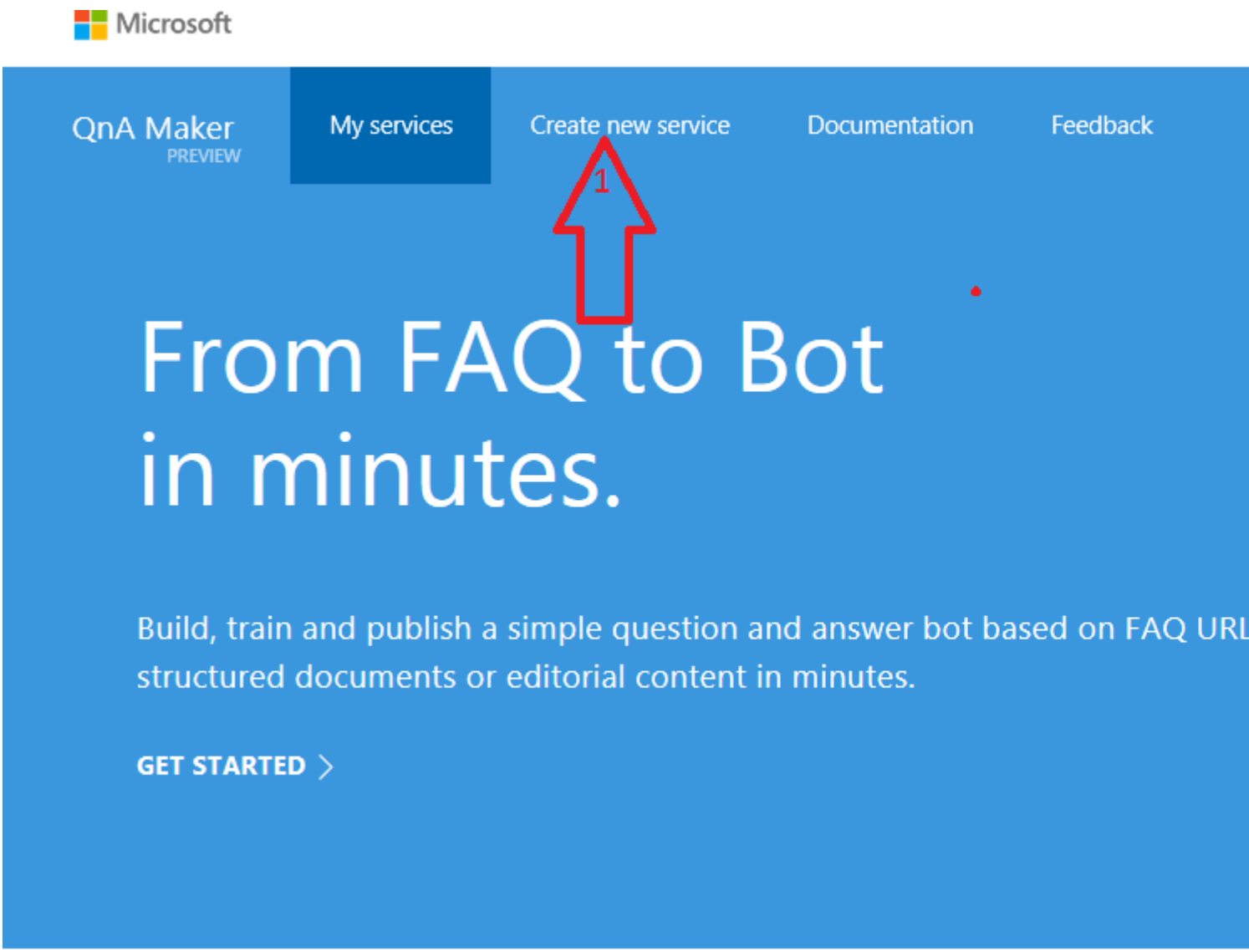
Introducción

QnA Maker es un servicio gratuito, fácil de usar, basado en API REST y basado en la web que entrena a AI para responder a las preguntas de los usuarios de una manera más natural y conversacional. Con la lógica de aprendizaje automático optimizada y la capacidad de integrar el procesamiento de lenguaje líder en la industria, QnA Maker destila datos semiestructurados como pares de preguntas y respuestas en respuestas distintas y útiles.

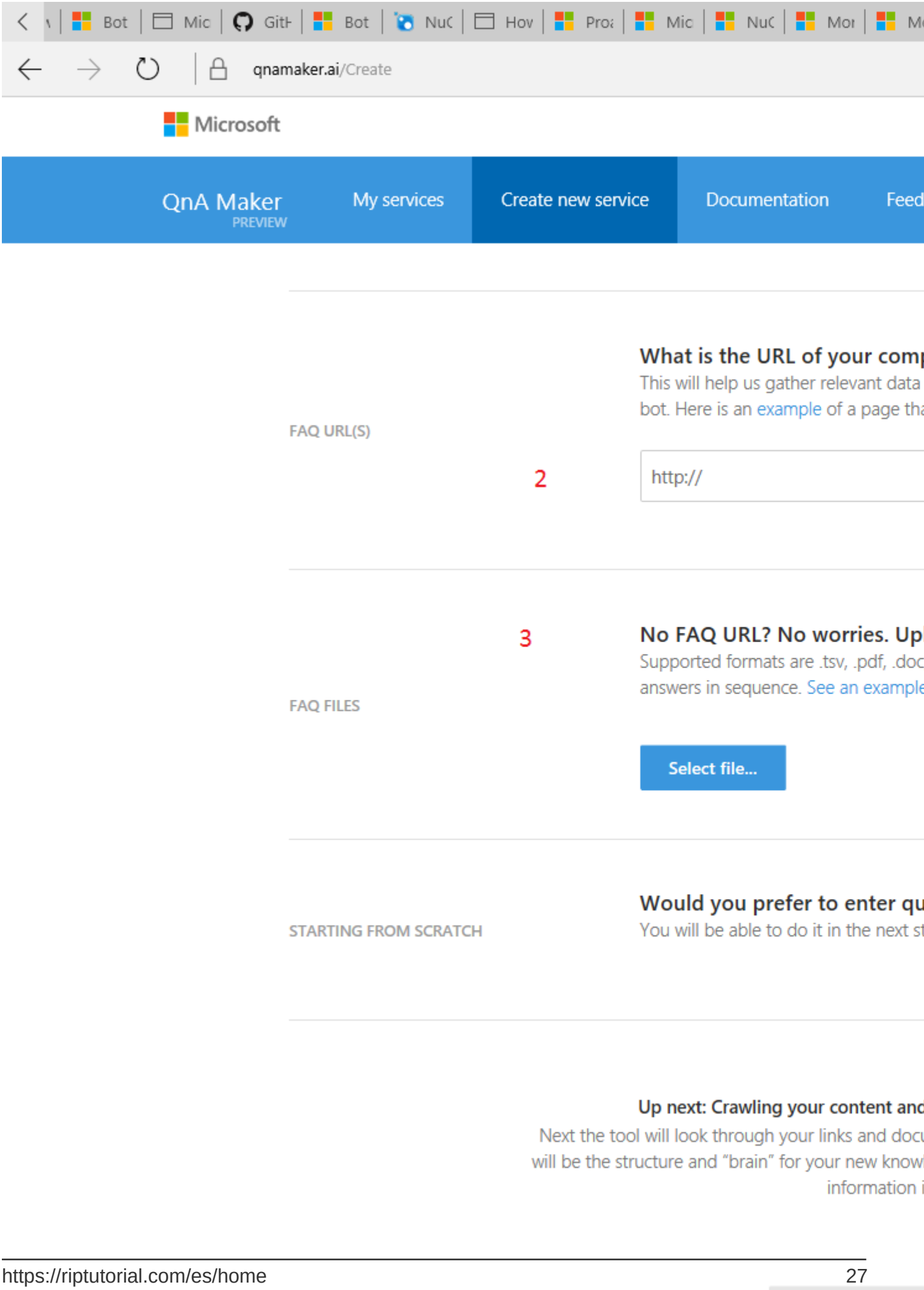
Examples

Creando nuestro propio servicio de QnA manualmente

Al proporcionar las credenciales de su cuenta de microsoft, puede autenticar y recibir claves de suscripción para comenzar con los servicios. Este documento describe los diversos flujos en la herramienta para crear su propia base de conocimientos.



QnA Maker trabaja en tres pasos: extracción, entrenamiento y publicación. Para empezar, aliméntelo desde URLs de preguntas frecuentes existentes a documentos y contenido editorial. He creado mi propia pregunta y respuestas manualmente.


qnamaker.ai/Create
Microsoft
QnA Maker
PREVIEW
My services
Create new service
Documentation
FAQ URL(S)
2
http://
3
FAQ FILES
Select file...
STARTING FROM SCRATCH
Up next: Crawling your content

y haciendo clic en "Comentarios" en la navegación superior.

Lea Comenzando con los servicios de QnA en línea:
https://riptutorial.com/es/botframework/topic/9520/comenzando-con-los-servicios-de-qna

Creditos

S. No	Capítulos	Contributors
1	Empezando con botframework	Community, Eric Dahlvang, Ezequiel Jadib, Mr. Kaffe Kup, Rajat Jain
2	Adición de procesamiento de lenguaje natural	Rajat Jain
3	Comenzando con el servicio de Azure Bot	Eric Dahlvang, Jyo Fanidam
4	Comenzando con los servicios de QnA	Jyo Fanidam