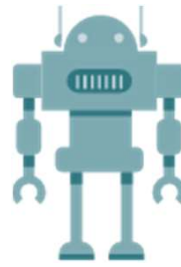


Building Intelligent Applications Using Bot Framework



Mithun Prasad, PhD
miprasad@Microsoft.com

What is a Bot?



What a Bot is Not

- Artificial Intelligence only
- Text interfaces only
- Natural Language Processing only



ryanv 8:26 AM
Cancel



Azure Bot Bot 8:26 AM
You have canceled the operation. What would you like to do next?



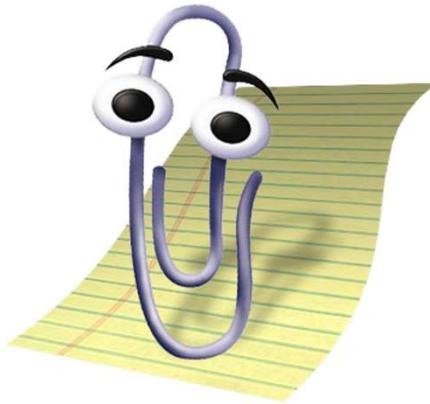
ryanv 8:26 AM
login



Azure Bot Bot 8:26 AM
Please select the subscription you want to work with:
1. Event Bots
2. QA Environment 1

What is a Bot?

Simply put, a bot is an application that performs an automated task. That's it



Principles of Bot Design



Bot Design

Assumptions

- Expect users to use the bot
- Users prefer the bot experience over alternative experiences (apps, websites, phone calls, etc)

Prioritize the right factors when designing your bot

Factors that do not guarantee a Bot's success

How “smart” the bot is

- Smarter bots may not guarantee happy users and adoption of your platform
- Many bots have little NLP capabilities

How much natural language the bot supports

- Bots can be great at conversations
- Bots can have a vocabulary and can even make great jokes
- Unless it addresses core problems, these capabilities may contribute very little to making your bot successful
- Some bots have no conversational capability. And in many cases, that's perfectly fine

Factors that do not guarantee a Bot's success

Voice

- Enabling bots for speech may not always lead to great user experiences
- Forcing users to use voice can result in a frustrating user experience
- Key considerations
 - Is voice the appropriate channel for the given problem?
 - Is the environment noisy?
 - Will voice convey the information that needs to be shared with the user?

Factors that do influence a Bot's success

A great user experience

Key considerations:

- Does the bot easily solve the user's problem with the minimal steps?
- Does the bot solve the user's problem better/easier/faster than any of the alternative experiences?
- Does the bot run on the devices and platforms the user cares about?
- Is the bot discoverable? Do the users naturally know what to do when using it?

Types of Bots



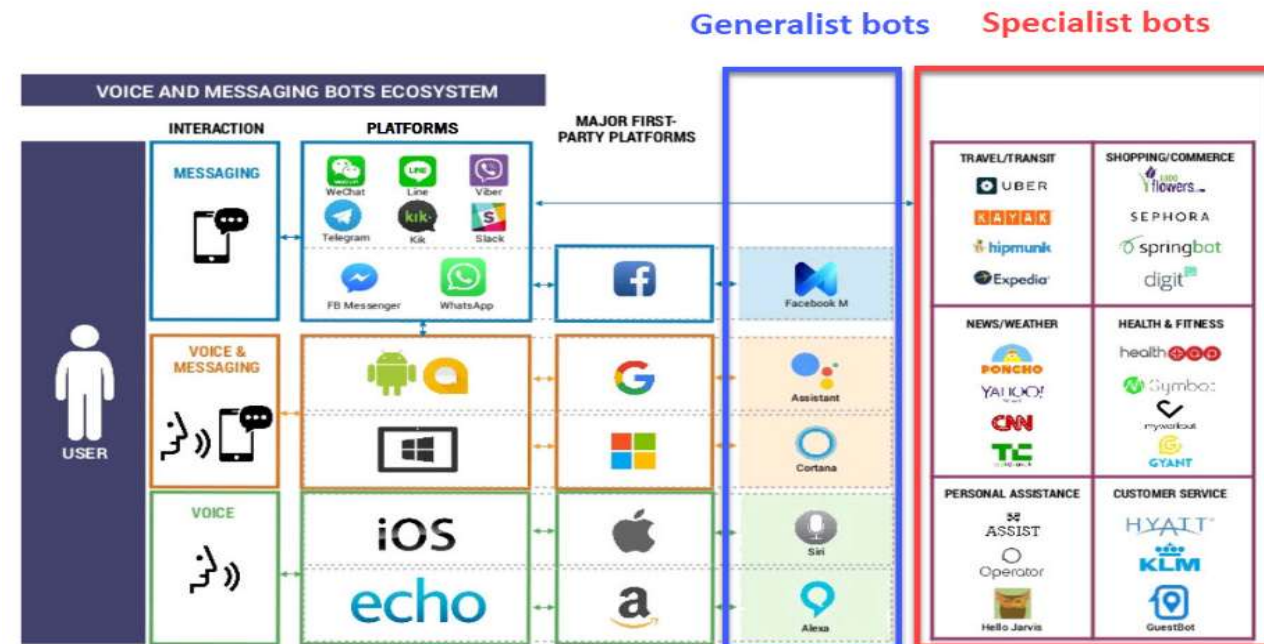
How Should We Categorize Bots?

Bot classification is arbitrary.

- Generalist bots vs Specialist bots
- Good bots vs Bad bots
- State related bots

Generalist Bots vs Specialist Bots

Recognize a primary market dynamic



Generalist Bots vs Specialist Bots

Generalist bots

- Can understand what you're asking
- No specific domain expertise to follow through with action
- In the enterprises, it is common where solutions are hard to generalize

Specialist bots

- Provides domain-specific knowledge necessary to accomplish tasks

Good Bots vs Bad Bots

Good Bots

| |
|---|
| Chatbots |
| Crawlers |
| Transactional bots |
| Informational bots |
| Entertainment bots: Art bots, Game bots |

Bad Bots

| |
|---------------|
| Hackers |
| Spammers |
| Scrapers |
| Impersonators |

Good Bots

| | |
|--------------------|---|
| Chatbots | Designed to carry forward conversations with humans and help in automation. |
| Informational bots | Obtain helpful information (for example, push notifications) and include things like breaking news stories. |
| Crawlers | <p>These bots run in the background, primarily fetch data from other APIs or websites, and are “well-behaved”</p> <p>Example: Pricing Assistant is bot that monitors ecommerce websites for price changes</p> |
| Transactional bots | <p>Act as agents on behalf of humans, and interact with external systems</p> <p>Example: A scheduler that finds meeting times for distributed teams</p> |

Bad Bots

| | |
|---------------|---|
| Hackers | Designed to distribute malware, deceive individual people, attack websites, and sometimes entire networks |
| Scrapers | Scraper bots steal content from other websites |
| Spammers | Post promotional content around the web, and eventually drive traffic to the spammer's website |
| Impersonators | Mimic natural user characteristics, making them hard to identify. |

State related Bots

Bots can also be split into the following categories:

- Stateless bots
- Semi-stateful bots
- Stateful bots

Stateless Bots

Example: Cleverbot (<https://cleverbot.io/>)

The services they offer are quite frivolous. Their main purpose is to bring entertainment value

- We can send some text to this bot and it will process and reply
- The conversation works but without reaching any definitive end goal
- The bot does not keep any memory of messages (i.e. it does not maintain the state of the conversation)
- Not capable of any learning

Semi-stateful Bots

Example: Automated Phone Menu System

- Limited ability to keep track of the conversation
- Unlike stateless bots, semi-stateful bots do not process each message as if it is coming from a brand new user
- Persist the basic details about the user (i.e. user ID, or phone number, etc.)
- The state they maintain is quite shallow
- Limit their state to a single conversation session with the user. If the user reaches their goal, the conversation ends

Stateful Bots

- Stateful bots keep the entire history of each conversation they had with the user
- The history of conversations can be used to adjust responses
- Query user for preferences
- Question and Answer

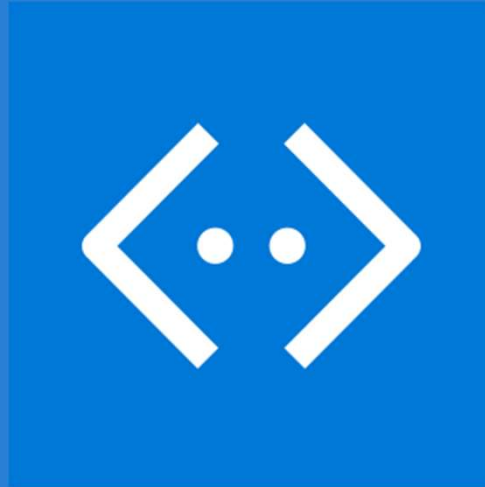
Microsoft Bot Framework





Bot Builder
SDK

Dev



Bot
Framework
Developer
Portal

Connect



Bot Directory

Publish

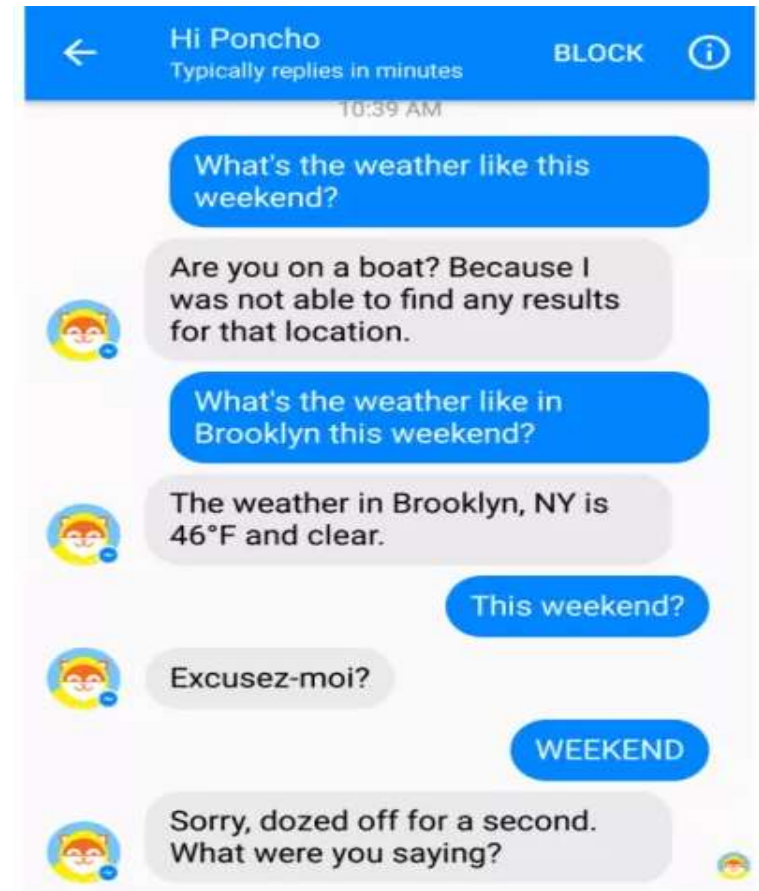
Bot Framework

Bot Framework

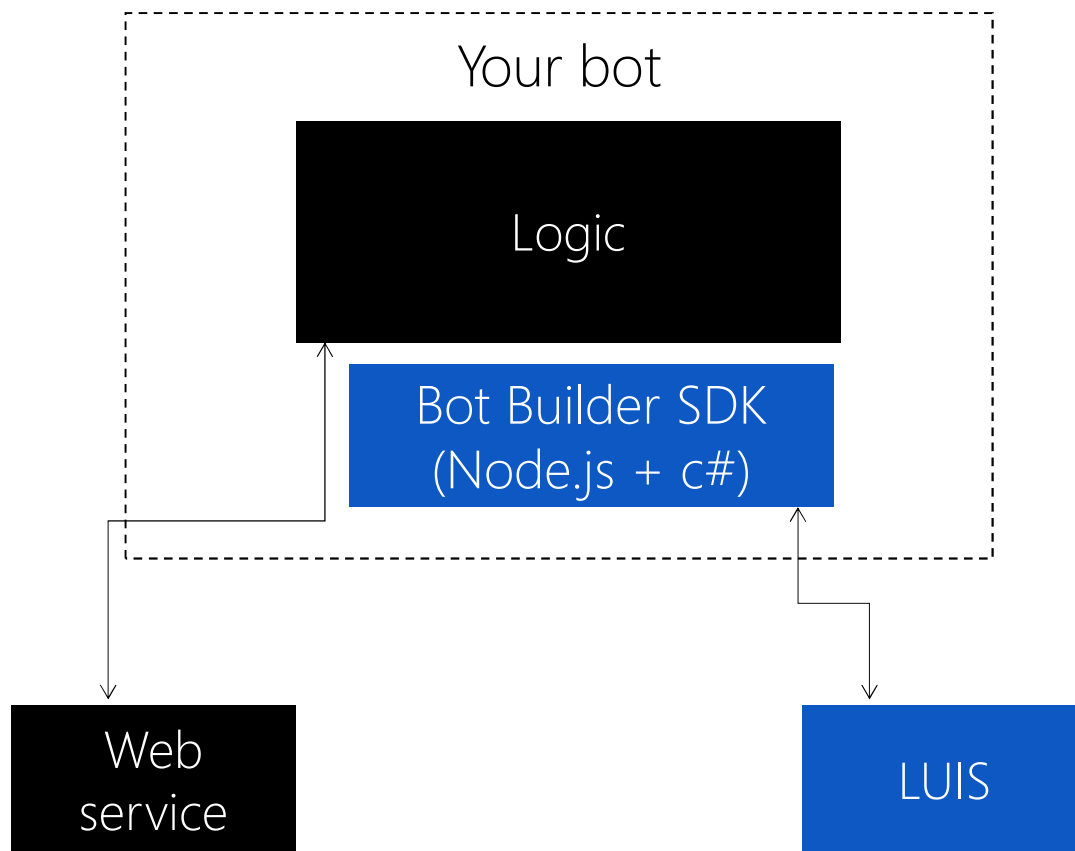
Bot Framework is a Microsoft-operated service and SDK

Bot Framework is one of many tools Microsoft offers for building a complete bot

Others include: LUIS, Speech APIs, Azure, more

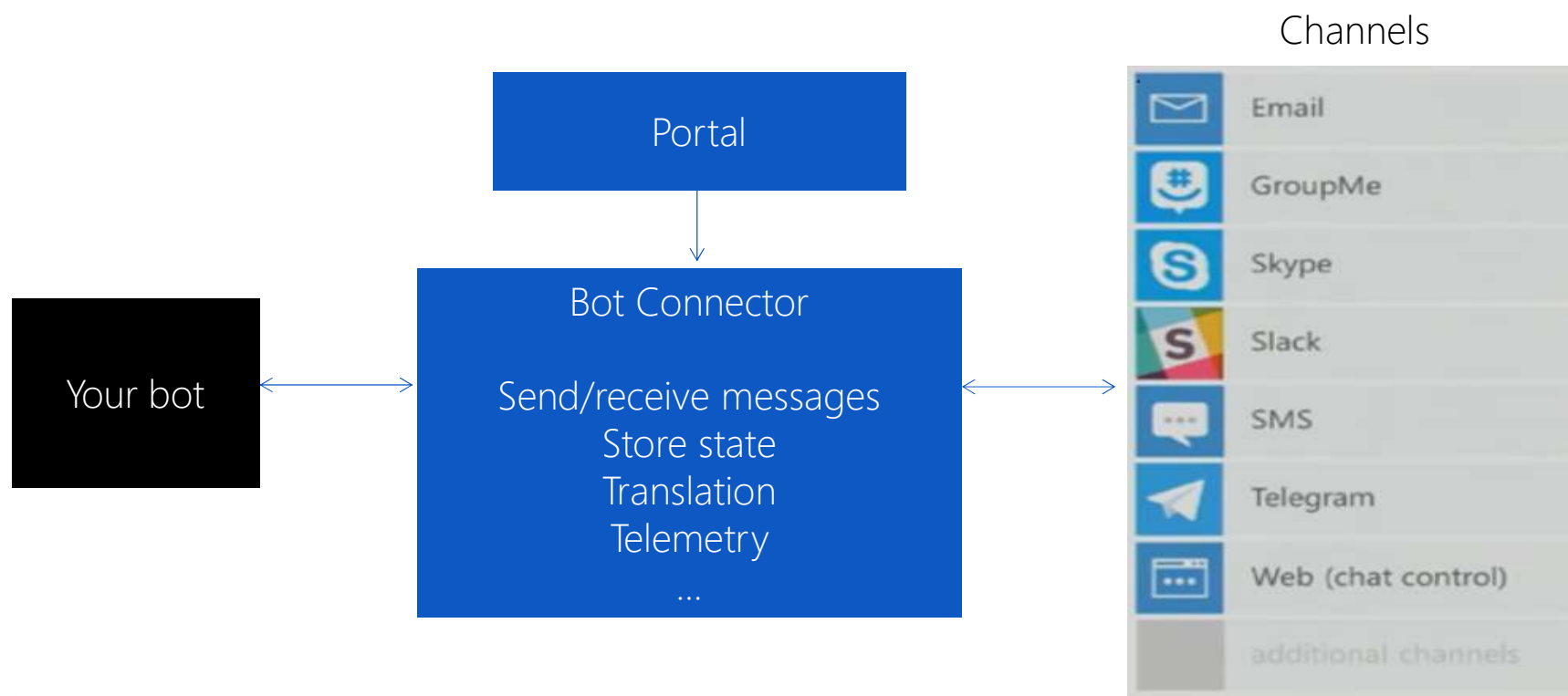


Bot Framework – Part 1



Bot Framework – Part 2

Bot Connector



Bot Connector Message

Your bot

Bot
Connector

```
{  
  "type": "message",  
  "id": "68YrxgtB53Y",  
  "conversationId": "DphpaFQrDuZDKyCez4AFGcT4vy5aQDjellLGJJB8v18MFtb",  
  "language": "en",  
  "text": "You can say \"/ order\" to order!",  
  "attachments": [],  
  "from": {  
    "name": "+12065551212",  
    "channelId": "sms",  
    "address": "+12065551212",  
    "id": "Ro52hKN287",  
    "isBot": false  
  },  
  "channelData": { SMS data here },  
  ...  
}
```

MessagesController

MessagesController.cs

```
1  [BotAuthentication]
2  public class MessagesController : ApiController
3  {
4      /// <summary>
5      /// POST: api/Messages
6      /// Receive a message from a user and reply to it
7      /// </summary>
8      public async Task<HttpResponseMessage> Post([FromBody]Activity activity)
9      {
10         if (activity.Type == ActivityTypes.Message)
11         {
12             await Conversation.SendAsync(activity, () => new Dialogs.RootDialog());
13         }
14
15         var response = Request.CreateResponse(HttpStatusCode.OK);
16         return response;
17     }
18 }
19
```

RootDialog

RootDialog.cs

```
1  [Serializable]
2  public class RootDialog : IDialog<object>
3  {
4      public Task StartAsync(IDialogContext context)
5      {
6          context.Wait(MessageReceivedAsync);
7
8          return Task.CompletedTask;
9      }
10
11     private async Task MessageReceivedAsync(IDialogContext context, IAwaitable<object> result)
12     {
13         await context.PostAsync(@"Hello, Bots!");
14
15         context.Wait(MessageReceivedAsync);
16     }
17 }
18
```

Key Concepts



Key Concepts

Bot Connector service

- The Bot Connector service enables your bot to exchange messages with channels configured in the [Bot Framework Portal](#)
- It uses industry-standard REST and JSON over HTTPS

Client libraries

- The Bot Framework provides client libraries (Bot Builder SDK) that can be used to build bots in either C# or Node.js. The SDK provides support for:
 1. Dialogs that encapsulate conversational logic
 2. Built-in prompts for Yes/No
 3. Powerful AI frameworks such as [LUIS](#), and more

Key Concepts

Channel

- A channel is the connection between the Bot Framework and communication apps such as:
 - Skype
 - Slack
 - Facebook Messenger
 - Office 365 mail
 - Others
- The Developer Portal is used to configure the channel
- Skype channel is the default channel

Key Concepts

Activity

- Bot Connector service exchanges information between bot and channel (user) by passing an Activity object

| Activity type | Description |
|-----------------------|---|
| message | Represents a communication between bot and user. |
| conversationUpdate | Indicates that the bot was added to a conversation, other members were added to or removed from the conversation, or conversation metadata has changed. |
| contactRelationUpdate | Indicates that the bot was added or removed from a user's contact list. |
| typing | Indicates that the user or bot is compiling a response. |
| ping | Represents an attempt to determine whether a bot's endpoint is accessible. |
| deleteUserData | Indicates to a bot that a user has requested that the bot delete any user data it may have stored. |
| endOfConversation | Indicates the end of a conversation. |

Key Concepts

State Data

State data can be used for many purposes:

- Determine where a prior conversation left off
- Greet a returning user by name
- With user's preferences, you can customize the conversation the next time you chat:
 - Example 1: Alert a user to a news article about a topic that is of interest
 - Example 2: Alert a user when an appointment becomes available

Bot State Service

- Enables your bot to store and retrieve state data that is associated with a user, a conversation, or a specific user within the context of a specific conversation
- Uses industry-standard REST and JSON over HTTPS
- Enables authentication with JWT Bearer tokens

Key Concepts

Authentication

Both the Bot Connector service and the Bot State service enable authentication with JWT Bearer tokens

JWT Bearer tokens

JSON Web Token (JWT) is an open standard that defines a compact and self-contained way for securely transmitting information between parties as a JSON object

This information can be verified and trusted because it is digitally signed

JWT Bearer Tokens

<https://jwt.io/>

Encoded

eyJhbGciOiJIUzI1NiIsInR5cCI6
IkpXVCJ9.eyJzdWIiOiIxMjM0NTY
3ODkwIiwibmFtZSI6IkpvaG4gRG9
lIiwiaWF0IjoiNjEwMDAwMDA0IiwiaW

Decoded

HEADER:

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD:

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "admin": true
}
```

VERIFY SIGNATURE

```
HMACSHA256(
    base64UrlEncode(header) + "." +
    base64UrlEncode(payload),
    secret
)
```

Secure your Bot

Use HTTPS and Bot Framework Authentication

To ensure that the bot's endpoint can only be accessed by the Bot Framework Connector, configure the bot's endpoint to use only HTTPS and enable Bot Framework authentication by registering your bot to acquire its app Id and password

Configure authentication

After the bot is registered, specify its App Id and password in your bot's web.config file.

```
<appSettings>  
  <add key="MicrosoftAppId" value="_appIdValue_" />  
  <add key="MicrosoftAppPassword" value="_passwordValue_" />  
</appSettings>
```

Secure your Bot

Authentication Credentials using Bot Builder SDK for .NET

To use the authentication credentials that are stored in the web.config file, specify the [BotAuthentication] attribute with no parameters

```
[BotAuthentication]  
public class MessagesController : ApiController  
{  
}
```

Channel-specific
Functionality



Channel-specific Functionality

- Some channels provide features that cannot be implemented using only message text and attachments
- To implement channel-specific functionality, you can pass native metadata to a channel in the Activity object's *ChannelData* property
- For example, you can instruct Office365 to send an email

Channel-specific Functionality

| | |
|----------|--|
| Email | Send and receive an email that contains body, subject, and importance metadata |
| Facebook | Send Facebook notifications natively |
| Telegram | Perform Telegram-specific actions, such as sharing a voice memo or a sticker |
| Kik | Send and receive native Kik messages |

Custom Email Message

| Property | Description |
|------------|--|
| htmlBody | An HTML document that specifies the body of the email message. See the channel's documentation for information about supported HTML elements and attributes. |
| importance | The email's importance level. Valid values are high, normal, and low. The default value is normal. |
| subject | The email's subject. See the channel's documentation for information about field requirements. |

```
"channelData": {  
  "htmlBody" : "<html><body>This is the email body!</body></html>",  
  "subject": "This is the email subject",  
  "importance": "high"  
}
```

Custom Telegram Message

| Property | Description |
|------------|---|
| method | The Telegram Bot API method to call. |
| parameters | The parameters of the specified method. |

```
"channelData": {  
  "method": "sendSticker",  
  "parameters": {  
    "sticker": {  
      "url": "https://domain.com/path/gif",  
      "mediaType": "image/gif",  
    }  
  }  
}
```

Facebook Notification

| Property | Description |
|-------------------|--|
| notification_type | The type of notification (e.g., REGULAR, SILENT_PUSH, NO_PUSH). |
| attachment | An attachment that specifies an image, video, or other multimedia type, or a templated attachment such as a receipt. |

```
"channelData": {  
  "notification_type": "NO_PUSH",  
  "attachment": {  
    "type": "template"  
    "payload": {  
      "template_type": "receipt",  
      ...  
    }  
  }  
}
```

Registration and
making your bots
available



Developer Portal: What Registration Does for You

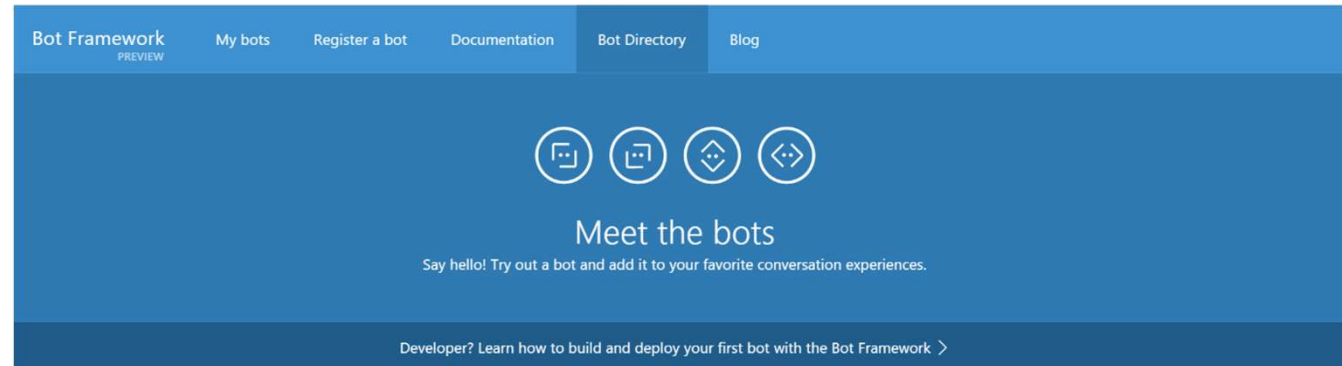
Your bot's web
service in the
cloud



MS Bot
Framework
Connector



Bot Directory


















Public Directory of Bot Framework Bots

<https://bots.botframework.com/>

- Discover, try, and add bots from here with no added configuration
- Bots are public at developer discretion; must be reviewed
- Searchable here

Featured Bots



| | | | | | |
|---|--|---|--|---|--|
|  | Skyscanner Skyscanner The world's travel search engine |  | AzureBot Microsoft Be more productive with your own Microsoft Azure subscriptions using |  | Spock Massively Chat with Spock, second in command of the USS Enterprise, to |
|  | Hipmunk Hipmunk, Inc. Hi, hi! I'm a super-powered travel assistant Hipmunk who can dig |  | Jessie Humani PullString, Inc. Jessie Humani's life is in your hands. Will you help her make some |  | SI x GameOn ... GameOn Technology Love the NFL? Add the SI x GameOn NFL bot to get unparalleled |
|  | StubHub StubHub Inc. Finding tickets to sports, concerts and theater as easy as chatting with |  | Mica, the Hips... LIKE A HIPSTER Hi, I'm Mica, your local hipster adviser. I help you discover the best |  | Baymax TrueCare24 Your personal healthcare companion. The service is available |
|  | MemeBot Fernando Galván Medina A simple bot that will assist you in your daily meme creation. You can |  | Cardea Powered by RingMD Meet Cardea, your personal medical aid. She will answer your health |  | Solitaire Card Lion Solitaire is the most popular card game in the world. |
|  | UNO Card Lion UNO world famous card game bot |  | Trivia Master Microsoft A fiendish quiz hosted by Professor T. Master. |  | RemitRadar RemitRadar Artificial Intelligence remittance assistant chatbot. Helps you to find |

Skyscanner

Bot Framework
PREVIEW

[My bots](#)

[Register a bot](#)

[Documentation](#)

[Bot Directory](#)

[Blog](#)



Skyscanner

Skyscanner | #travel, #flights, #inspiration

Overview

The world's travel search engine

[Privacy statement](#) | [Publisher email](#) | [Bot website](#) | [Report abuse](#)

Say hello

Add this bot to your favorite conversation experiences.



Skype

 Add to Skype

Skyscanner



Skyscanner



Skype Certified 

Skyscanner



Add to Contacts



 Home


 Contacts

Recent


 **Skyscanner** 
What's the start date...



 History



 **Skyscanner** 
Online


Tip of the day: at any point you can say 'help' for a more detailed explanation.  10:19 am



Today

 Hello!

  Open

  Open

 From which city will you start your trip?

  Open

Hi 10:20 am

10:20 am

10:20 am




I want to travel to Denmark 10:21 am

10:21 am

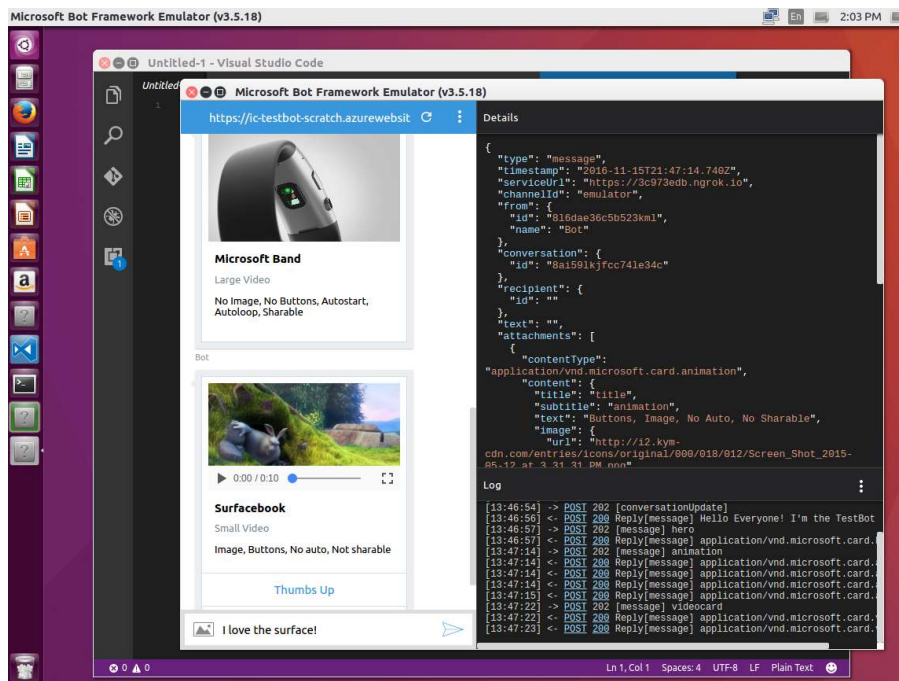
Los Angeles 10:21 am

via Skype

Type a message here

The BF Emulator to Test a Bot












- Support for Mac, Linux and Windows
- Send requests and receive responses to/from your bot endpoint on localhost
- Inspect Json chat response


Adding a Channel

- Skype is added by default
- Instructions are laid out

Add another channel


| | |
|---|--------------------|
|  | Direct Line |
|  | Email |
|  | Facebook Messenger |
|  | GroupMe |
|  | Kik |
|  | Microsoft Teams |
|  | Slack |
|  | Telegram |
|  | Twilio (SMS) |



 Microsoft

Micheleen Harris ▾

Configure Slack



How to

^ Log in to Slack and create a Slack Application for your bot

<https://api.slack.com/applications/new>

Your Apps

Create New App

▽ Create application and set redirect URI

Register a Bot in The Developer's Portal



Bot Framework
PREVIEW

My bots

Register a bot

Documentation

Bot Directory

Blog

Tell us about your bot

Bot profile



Icon

[Upload custom icon](#)

30K max, png only

* Name ?

* Bot handle ?

* Description ?



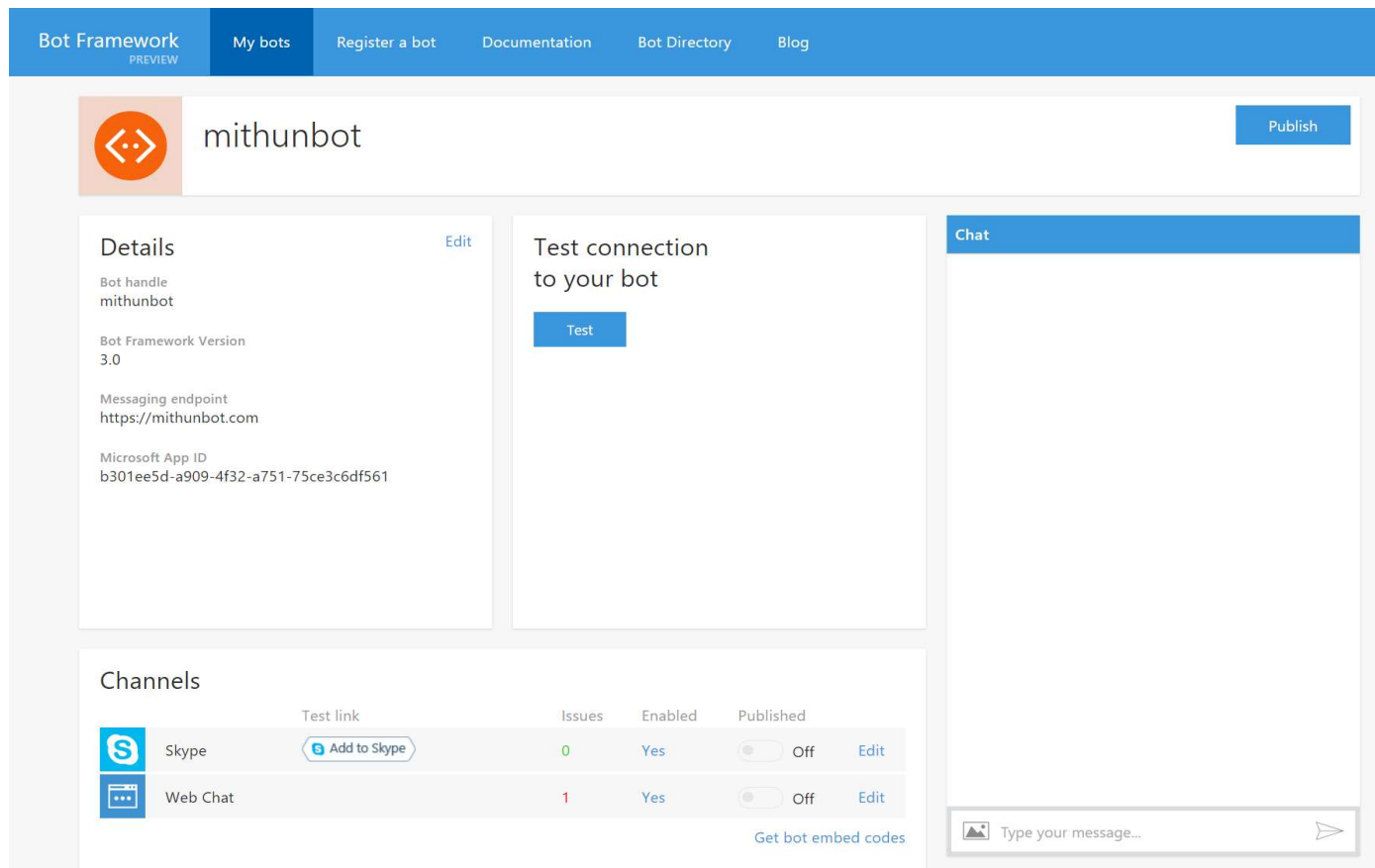

Register a Bot in The Developer's Portal

The screenshot shows the Microsoft Bot Framework Developer's Portal interface for a bot named "mithunbot". The top navigation bar includes links for "Bot Framework PREVIEW", "My bots", "Register a bot", "Documentation", "Bot Directory", and "Blog".

The main content area is divided into several sections:

- Bot Header:** Displays the bot's icon (an orange circle with a white double arrow) and the name "mithunbot". A "Publish" button is located in the top right corner.
- Details Panel:** Contains the following information:
 - Bot handle: mithunbot
 - Bot Framework Version: 3.0
 - Messaging endpoint: https://mithunbot.com
 - Microsoft App ID: b301ee5d-a909-4f32-a751-75ce3c6df561An "Edit" link is visible in the top right of this panel.
- Test connection to your bot:** A section with a "Test" button.
- Channels:** A table listing the bot's configured channels.
- Chat:** A large text area for testing the bot's responses, with a "Type your message..." input field and a send button at the bottom.

The "Channels" table has the following data:

| | Test link | Issues | Enabled | Published | |
|--|------------------------------|--------|---------|------------------------------|----------------------|
|  Skype | Add to Skype | 0 | Yes | <input type="checkbox"/> Off | Edit |
|  Web Chat | | 1 | Yes | <input type="checkbox"/> Off | Edit |

Below the table, there is a link to "Get bot embed codes".

The Microsoft logo is visible in the bottom left corner of the image.



Register a Bot in The Developer's Portal

Channels

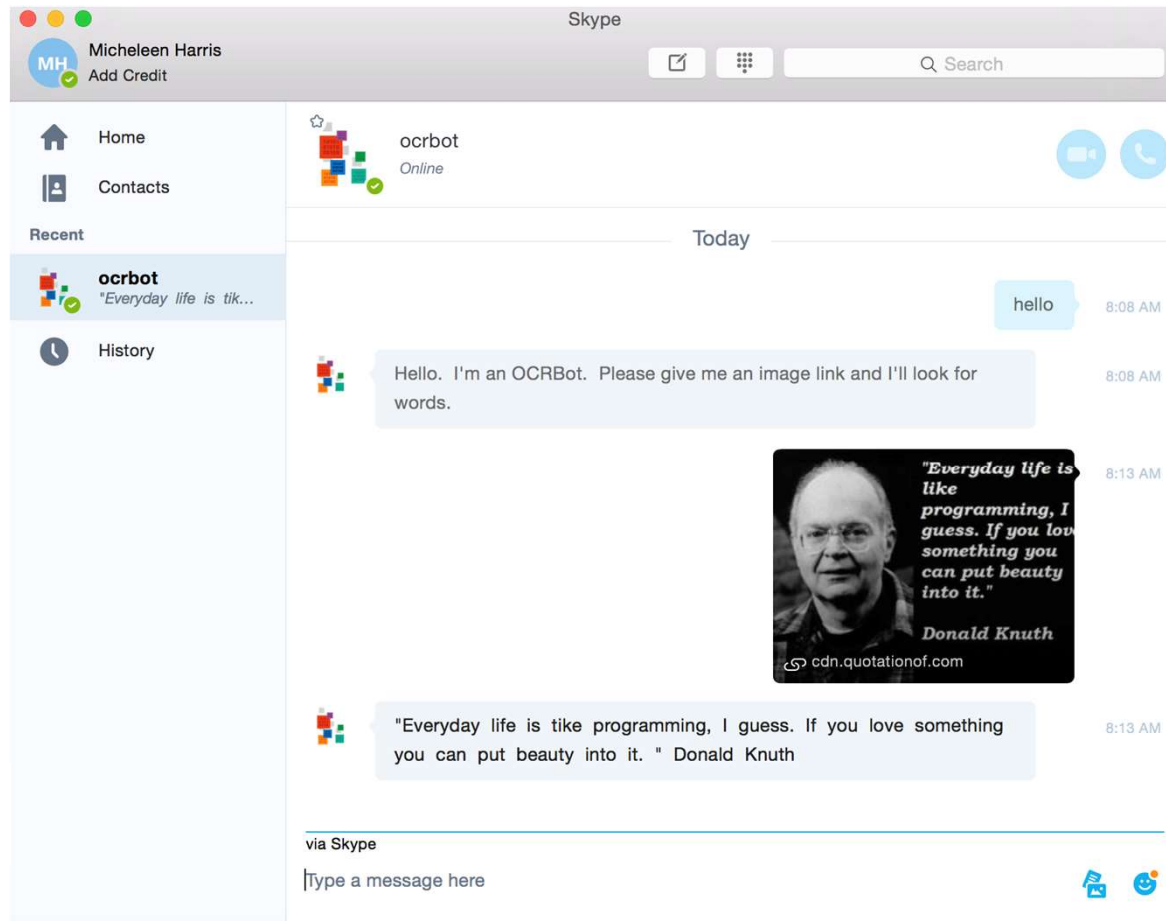
| | | Test link | Issues | Enabled | Published | |
|---|----------|------------------------------|--------|---------|------------------------------|----------------------|
|  | Skype | Add to Skype | 0 | Yes | <input type="checkbox"/> Off | Edit |
|  | Web Chat | | 1 | Yes | <input type="checkbox"/> Off | Edit |

[Get bot embed codes](#)

Add another channel

| | | |
|---|--------------------|---------------------|
|  | Direct Line | Add |
|  | Email | Add |
|  | Facebook Messenger | Add |
|  | GroupMe | Add |
|  | Kik | Add |
|  | Microsoft Teams | Add |
|  | Slack | Add |
|  | Telegram | Add |
|  | Twilio (SMS) | Add |

Skype Channel Example



Slack Channel Example



bot-education ▾

● michhar

CHANNELS (2) +

general

random

DIRECT MESSAGES (3) +

♥ slackbot

● michhar (you)

+ Invite people

#random

2 | 0 | Non-work banter and water cooler conversation

📞 ⚙️ 📄

🔍 Search

@ ☆ ⋮

ocrbot BOT 5:07 AM

You never fail until you stop trying. Albert Einstein

michhar 7:47 AM

<http://www.desicommments.com/dc3/03/231578/231578.jpg> (99KB) ▾

ocrbot BOT 7:47 AM

A PERSON WHO NEVER MADE A MISTAKE NEVER TRIED ANYTHING NEW. -ALBERT EINSTEIN

+ Message #random

😊

LAB:

BotsTennisLab



LAB:

BotsPublishingAnd
RegisteringLab

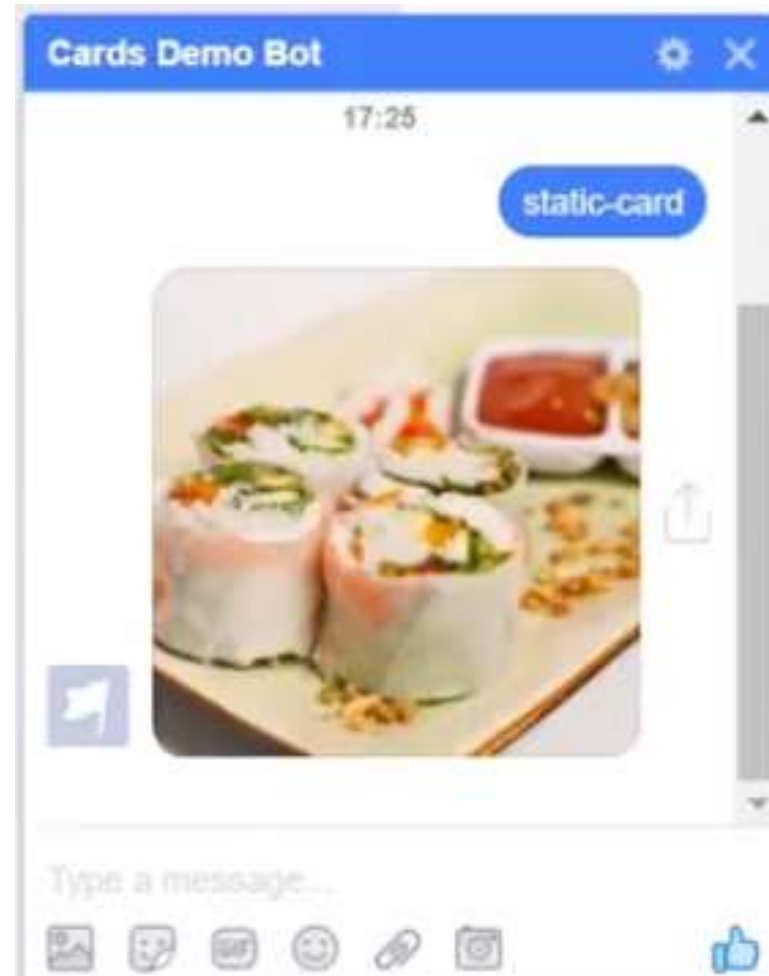


Messages and Activities



Rich Cards

- Provide a title
- A subtitle
- A button to take some action
- And a picture

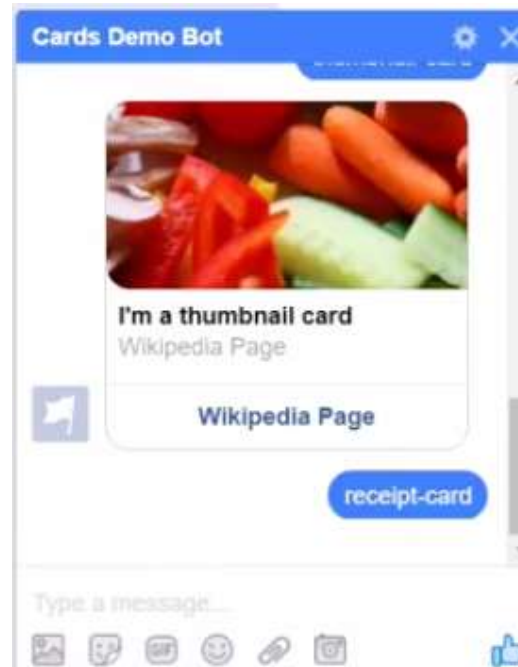
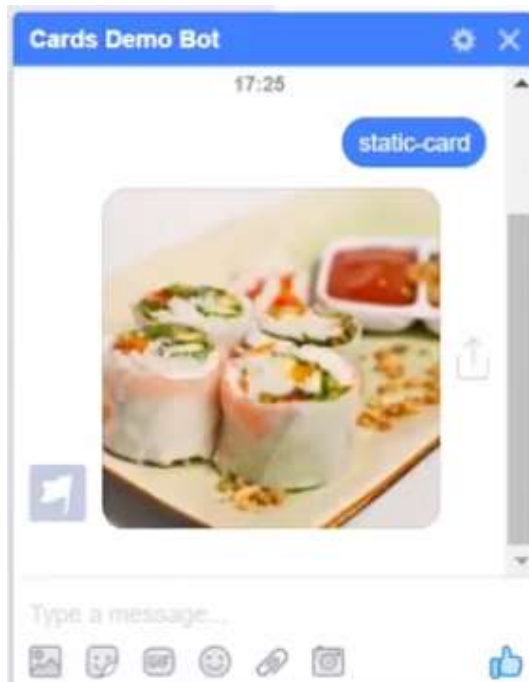


Rich Cards:

Types of Rich Cards

| Card type | Description |
|---------------|---|
| AdaptiveCard | Contains any combination of text, speech, images, buttons, and input fields |
| AnimationCard | Can play animated GIFs or short videos |
| AudioCard | Can play an audio file |
| HeroCard | Typically contains a single large image, one or more buttons, and text |
| ThumbnailCard | Typically contains a single thumbnail image, one or more buttons, and text |
| ReceiptCard | Typically contains the list of items to include on the receipt, tax and total information, and other text |
| SignInCard | Typically contains text and one or more buttons that the user can click to initiate the sign-in process |
| VideoCard | A card that can play videos |

Rich Cards: Examples



Rich Cards:

Process Events within Rich Cards

| Property | Type | Description |
|----------|--------|---|
| Type | string | type of action (one of the values specified in the table below) |
| Title | string | title of the button |
| Image | string | image URL for the button |
| Value | String | value needed to perform the specified type of action |

Creating Cards - Thumbnail

- The Thumbnail card typically contains a single thumbnail image, one or more buttons, and text.

```
var thumbnail = new ThumbnailCard();
thumbnail.Title = profile.Login;
thumbnail.Images = new[] { new CardImage(profile.AvatarUrl) };

if (!string.IsNullOrEmpty(profile.Name)) thumbnail.Subtitle = profile.Name;

string text = string.Empty;
if (!string.IsNullOrEmpty(profile.Company)) text += profile.Company + " \n";
if (!string.IsNullOrEmpty(profile.Email)) text += profile.Email + " \n";
if (!string.IsNullOrEmpty(profile.Bio)) text += profile.Bio;

thumbnail.Text = text;

thumbnail.Buttons = new[] { new CardAction(ActionTypes.OpenUrl, @"Click to view", value: profile.HtmlUrl) };
```

Creating Cards – Hero Card

- The Hero card typically contains a single large image, one or more buttons, and text.

```
Activity reply = activity.CreateReply("Here's a **hero**");

var heroCard = new HeroCard()
{
    Title = "I'm a hero card",
    Subtitle = "Robin hearts Tachikoma",
    Images = new List<CardImage> {
        new CardImage(url: "http://robinosborne.co.uk")
    },
    Buttons = new List<CardAction> {
        new CardAction()
        {
            Value = "https://robinosborne.co.uk/?s=bot",
            Type = "openUrl",
            Title = "Rob's bots"
        }
    }
};

reply.Attachments = new List<Attachment> {
    heroCard.ToAttachment()
};
```

Creating Cards – Sign In Card

- When the card action is set to the type of "signin", it can initiate an oauth flow; the user can be made to leave the conversation to authenticate with a third party system.

```
Activity reply = activity.CreateReply("Here's a **signin**");

var signinCard = new SigninCard()
{
    Text = "Authorisation needed",
    Buttons = new List<CardAction> {
        new CardAction()
        {
            Value = "https://robinosborne.co.uk/?s=bot",
            Type = "signin",
            Title = "Facebook OAuth",
            Image = "https://cdn1.iconfinder.com/data/icons/logotypes"
        },
        new CardAction()
        {
            Value = "https://robinosborne.co.uk/?s=bot",
            Type = "signin",
            Title = "Google",
            Image = "http://images.dailytech.com/nimage/G_is_For_Google"
        }
    }
};
```

Intercept Messages

- The middleware functionality in the Bot Builder SDK enables your bot to intercept all messages that are exchanged between user and bot
- For message(s) that are intercepted,
 - save the message to a data store
 - write to a conversation log
 - inspect the message in some way and take whatever action your code specifies

Proactive Messages

- Typically, each message that a bot sends to the user directly relates to the user's prior input
- In some cases, a bot may need to send the user a message that is not directly related to the current topic of conversation. These types of messages are called proactive messages

Proactive Messages

- Proactive messages can be useful in a variety of scenarios:
 - If a bot sets a timer or reminder, it will need to notify the user when the time arrives
 - If a bot receives a notification from an external system, it may need to communicate that information to the user immediately. For example, if the user wants the bot to monitor the price of a product, the bot will alert the user if it receives notification that the price of the product has dropped by 20%.

Global Message Handlers

- Users commonly attempt to access certain functionality within a bot by using keywords like "help", "cancel", or "start over"
- By implementing global message handlers, you can design your bot to gracefully handle such requests during different points of the conversation
- The handlers will examine user input for the keywords that you specify, such as "help", "cancel", or "start over", and respond appropriately



Do you confirm this order?

Yes

No

Proactive Messages:

Types of Proactive Messages

Ad hoc proactive message

- The bot simply interjects the message into the conversation whenever it is triggered
- The bot will not attempt to change the conversation in any way

Dialog-based proactive message

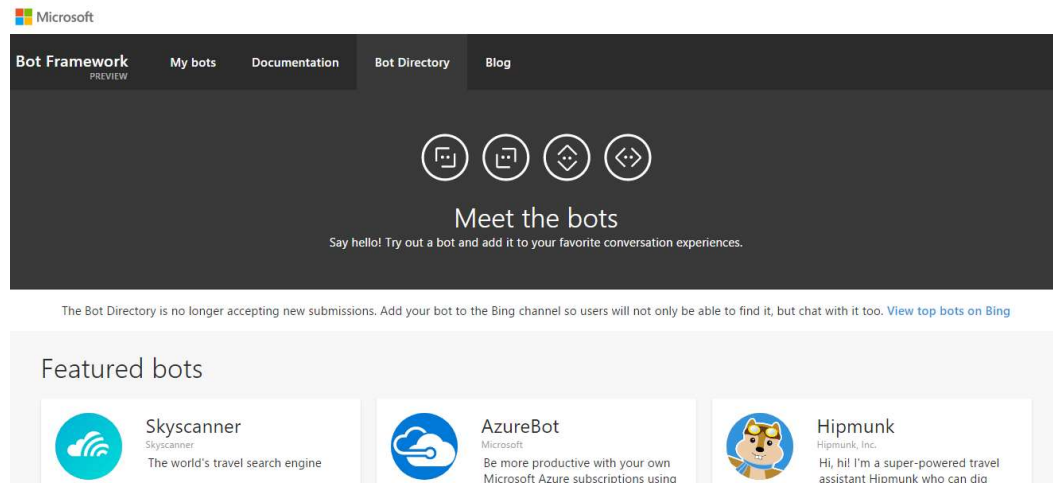
- The bot must identify the context of the existing conversation before interrupting a conversation
- Decide how (or if) it will resume that conversation after the message interrupts

Connecting to
Channels



Making Your Bot Available

- Bot Directory



- Bing channel
- Skype for Business
- Channels have their own way of making bots available

Third Party Channels

Example: Connect to Facebook Messenger

- On the Bot Framework Portal, you will need Page ID, App ID, App Secret, and Page Access Token values obtained from Facebook Messenger
- After the bot is finished, Facebook has its own review process for apps that are published to Messenger.
- The bot will be tested to ensure it is compliant with Facebook's Platform Policies



Enter your Facebook Messenger credentials

[Where do I find my Facebook Messenger credentials?](#)

Facebook Page ID

Facebook App ID

Facebook App Secret

Page Access Token

Connect to Channels (Facebook)

Page ID


- The bot is accessed through a Facebook Page
- Create a new Facebook Page or go to an existing Page
- Obtain the Page ID from Facebook Page's About page

Connect to Channels (Facebook)

Facebook app

- Create a new Facebook App on the Page and generate an App ID and App Secret for it
- Obtain the App ID and the App Secret

Dashboard



TestBot

This app is in development mode and can only be used by app admins, developers and testers

API Version v2.6

App ID

App Secret

Create a New App ID

Get started integrating Facebook into your app or website

Display Name

The name of your app or website

☐ No

Is this a test version of another app? [Learn More.](#)

Contact Email

Used for important communication about your app


Category

Choose a Category

By proceeding, you agree to the Facebook Platform Policies

Cancel

Create App ID



Connect to Channels (Facebook)

Enable messenger

- Enable Facebook Messenger in the new Facebook App



Product Setup

Facebook Login

The world's number one social login product.

[Get Started](#)

Audience Network

Monetize your mobile app or website with native ads from 3 million Facebook advertisers.

[Get Started](#)

Account Kit

Seamless account creation. No more passwords.

[Get Started](#)

Messenger

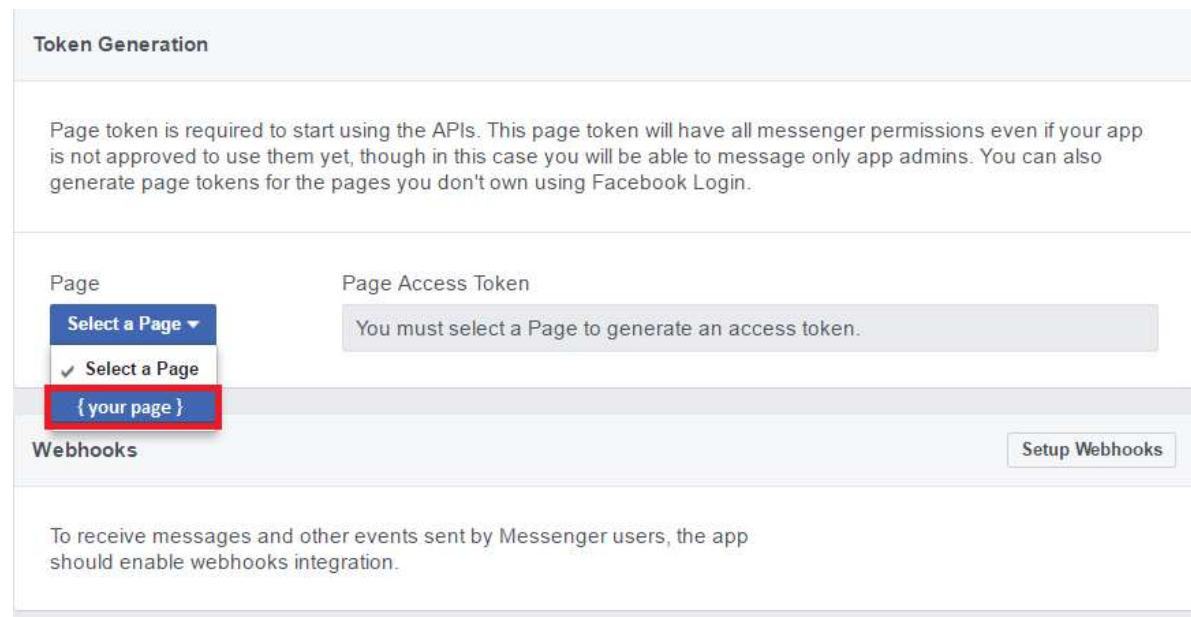
Customize the way you interact with people on Messenger.

[Get Started](#)

Connect to Channels (Facebook)

Generate a Page Access Token

- In the Token Generation panel of the Messenger section, select the target Page. A Page Access Token will be generated.
- Obtain the Page Access Token



The screenshot shows the 'Token Generation' panel in the Facebook developer console. It includes a text block explaining that a page token is required for API use and that it grants all messenger permissions. Below this, there are two main sections: 'Page' and 'Page Access Token'. The 'Page' section features a dropdown menu labeled 'Select a Page' with a checkmark and the text '{ your page }' highlighted in a red box. The 'Page Access Token' section contains a text input field with the placeholder text 'You must select a Page to generate an access token.' Below these sections is a 'Webhooks' section with a 'Setup Webhooks' button and a text block stating that webhooks integration should be enabled to receive messages and events.

Token Generation

Page token is required to start using the APIs. This page token will have all messenger permissions even if your app is not approved to use them yet, though in this case you will be able to message only app admins. You can also generate page tokens for the pages you don't own using Facebook Login.

Page

Select a Page ▼

✓ Select a Page

{ your page }

Page Access Token

You must select a Page to generate an access token.

Webhooks

Setup Webhooks

To receive messages and other events sent by Messenger users, the app should enable webhooks integration.

Connect to Channels (Facebook)

Enable webhooks

Click Set up Webhooks to forward messaging events from Facebook Messenger to the bot

Webhooks

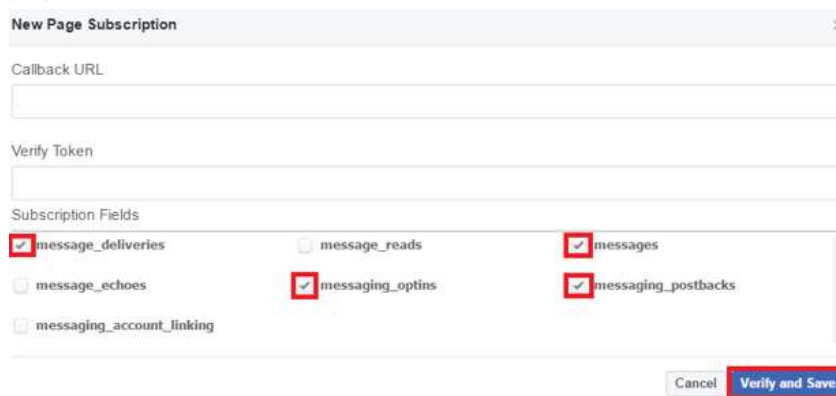
[Setup Webhooks](#)

To receive messages and other events sent by Messenger users, the app should enable webhooks integration.

Connect to Channels (Facebook)

Provide webhook callback URL and verify token

- In the Bot Framework Portal, open the bot, click the Channels tab, and then click Facebook Messenger.
- Copy the Callback URL and Verify Token values from the portal
- In Facebook Messenger, paste the Callback URL and Verify Token values
- Under Subscription Fields, select message_deliveries, messages, messaging_options, and messaging_postbacks



New Page Subscription

Callback URL

Verify Token

Subscription Fields

| | | |
|--|---|---|
| <input checked="" type="checkbox"/> message_deliveries | <input type="checkbox"/> message_reads | <input checked="" type="checkbox"/> messages |
| <input type="checkbox"/> message_echoes | <input checked="" type="checkbox"/> messaging_optsins | <input checked="" type="checkbox"/> messaging_postbacks |
| <input type="checkbox"/> messaging_account_linking | | |

Cancel Verify and Save

Callback URL and Verify Token for Facebook

What do I do with my Callback URL and Verify token?

Callback URL (Copy and paste in Facebook)

Verify Token (Copy and paste in Facebook)



LAB:

BotsRichUILab

