

Building a Webex Bot Quickly

Hands-On Demo

Phil Bellanti – Developer Advocate
Webex Cloud Platform
@PBellanti



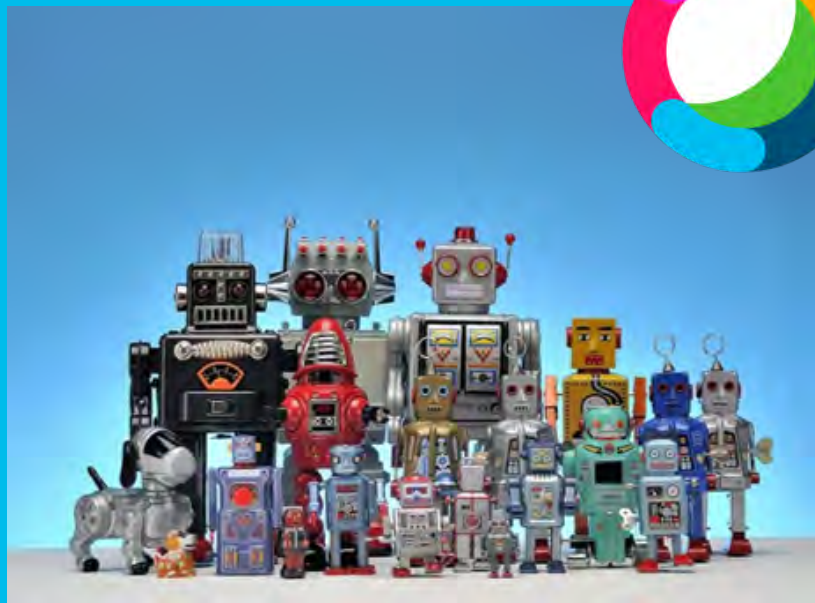


Agenda

- Intro to Webex Messaging Bots
- Getting Started
- Setting up a Local Web Sever
- Configure and Run the Bot App
- Interact with the Bot in Webex Teams and Inspect the Code
- Explore More

Intro to Webex Bots

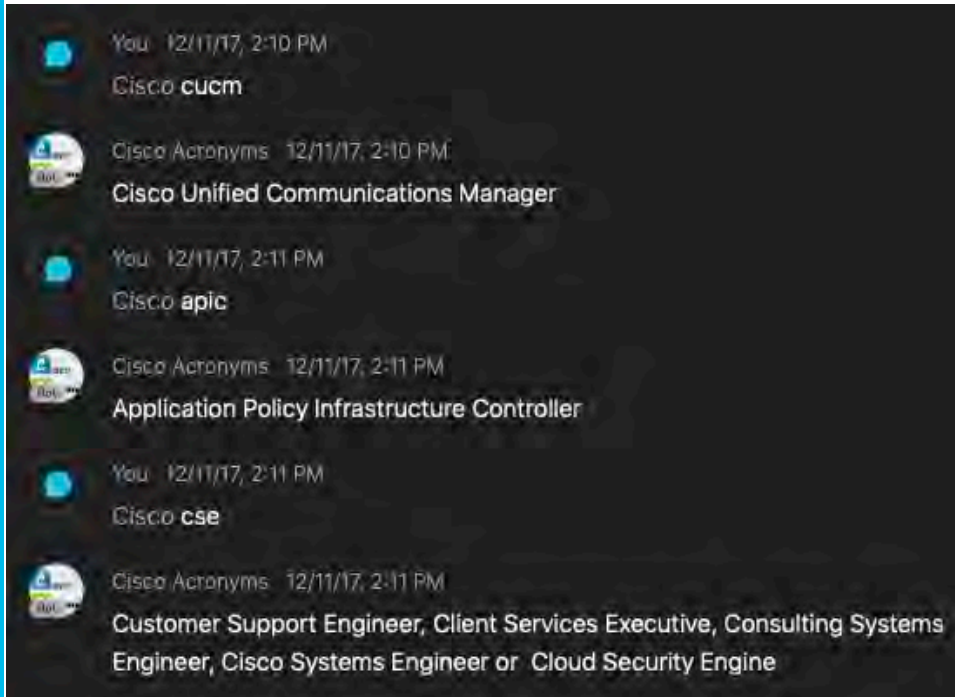
What are Messaging Bots?



- Messaging is big business
- Messaging bots are now becoming expected in the workplace to:
 - Automate tedious or complex tasks
 - Provide fast assistance in a human-relatable way
 - Connect all the services a user finds more important to streamline workflows

Webex Bots

- Exclusive to Webex Messaging
- Special “Bot” badge
- Functionality limited only by the imagination of the developer



Webex Messaging Bot Use Cases



Customer and
User Services



Meetings &
Scheduling



Content
Management



Task
Management

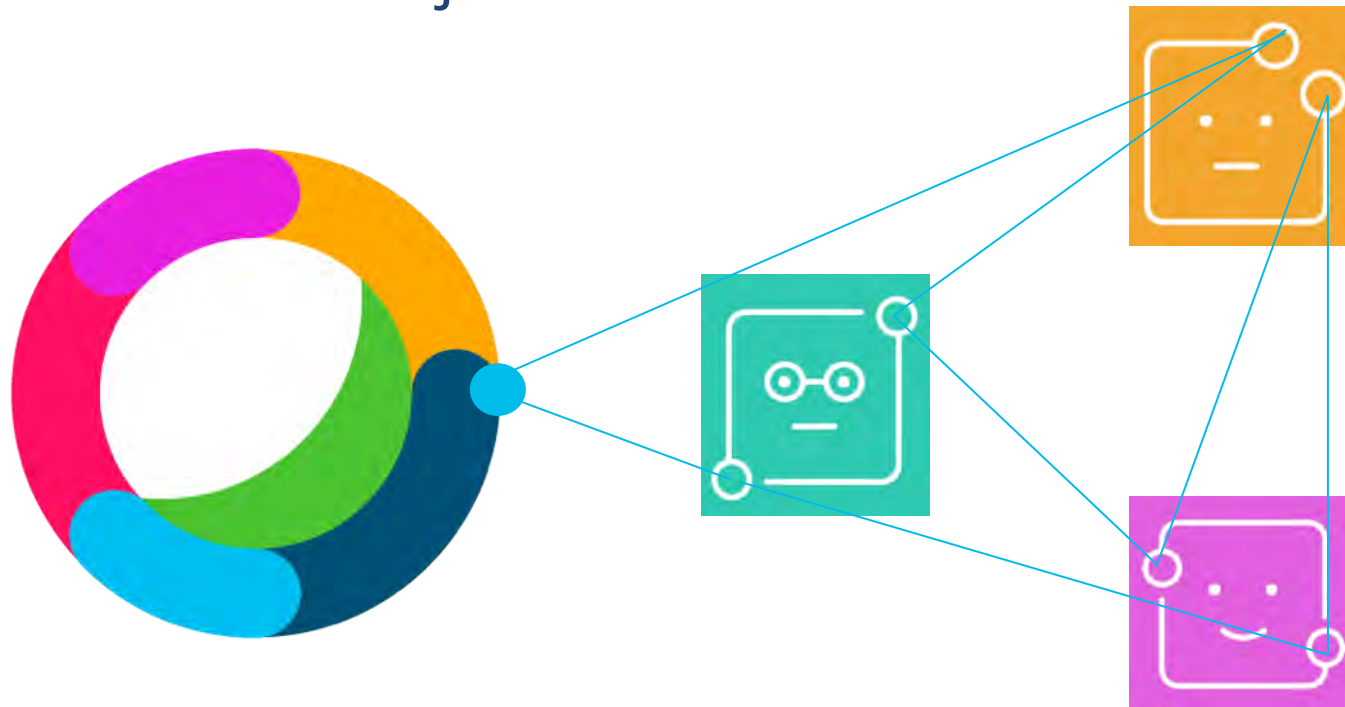


Polling



Notifications
& Alerts

Webex Node.js Bot Framework



Webex Node.js Bot Framework



<https://eurl.io/#BJ7gmISeU>

Getting Started

Installations

Node.js (version 8.0.0 or higher) *npm*, and *git* installed on your machine. A text editor is also recommended.

- Node.js: <https://nodejs.org/>
- npm: <https://www.npmjs.com/get-npm>
- git: <https://git-scm.com/>

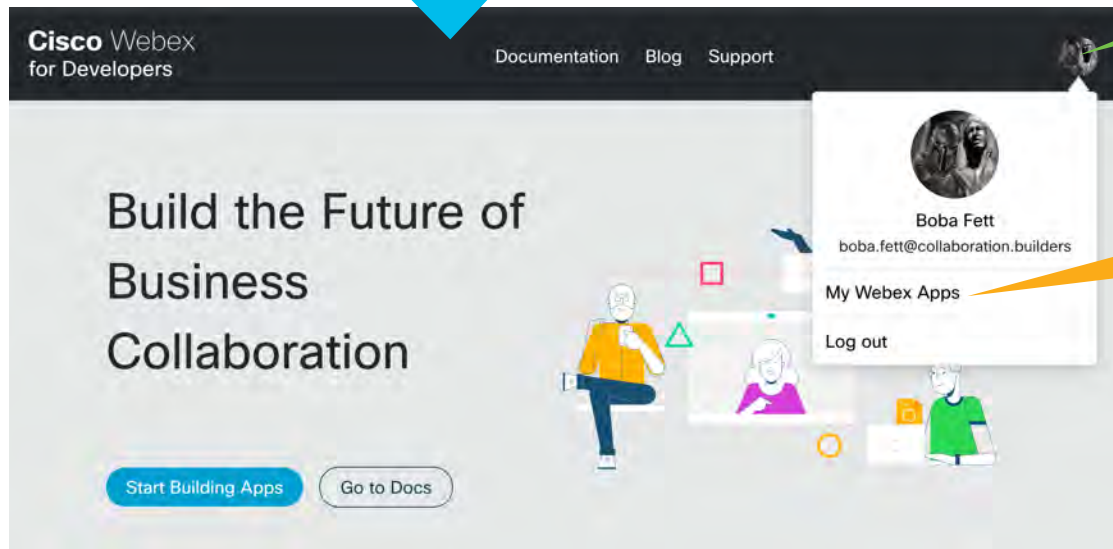


Create Bot Account

1. Log in to
developer.webex.com

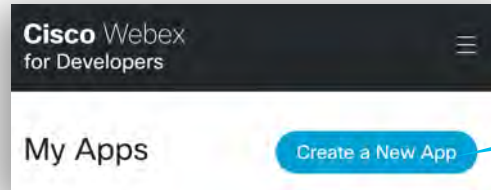
2. Click
avatar

3. Select “My Webex Apps”

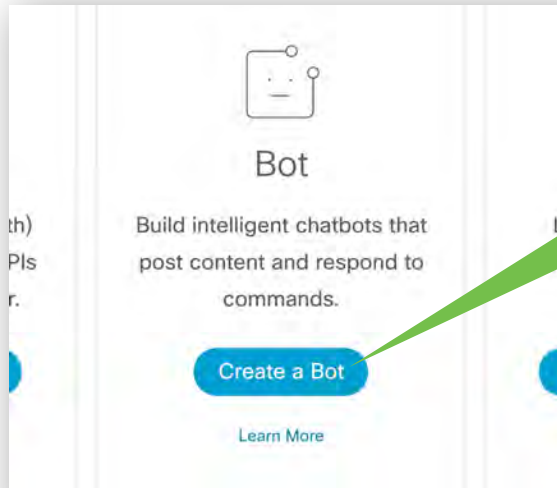


Create Bot Account (continued)

3. Provide the bot info



1. Click "Create a New App"



2. Click "Create a Bot"

New Bot

Bot Name*
Name of your bot in 100 characters or less.

Ex: Sample

Bot Username*
The username users will use to add your bot to a space. Cannot be changed later.

Ex: sparkybot @webex.bot

Icon*
Upload your own or select from our defaults. Must be exactly 512x512px in JPEG or PNG format.

Upload Default 1 Default 2 Default 3

Description*
Provide some details about what

Create Bot Account (continued)

Congratulations!

Your bot has been created. Use your Bot Access Token to setup your webhook and finish building your app. When you're ready, return to [submit your app](#) to Webex App Hub.

Node.js Bot Demo [Submit to Webex App Hub](#)

Bot's Access Token
Non-expiring (good for 100 years) access token for your bot.

NmVIOTg4MmQtNWl0NS00ZjJhLWlyMTgtODYzZjQwK([Copy](#)

Bot ID
Unique system generated ID for your bot.

Y2IzY29zcGFyazovL3VzL0FQUEXJQ0FUSU9OLzIzODY([Copy](#)

Bot Name*
Name of your bot in 100 characters or less.

Node.js Bot Demo [Edit](#)

Bot Username*
The username users will use to add your bot to a space. Cannot be changed later.

node.demo.bot.4321@webex.bot

Copy bot access token and keep in safe place

Copy bot username and keep it with the access token

Setting up a Local Web Server

Setting up nGrok on Your Machine

1. Download and setup nGrok here: <https://ngrok.com/>
2. After unzipping nGrok, from the **same** directory as the nGrok executable, execute this command line:

```
$ ./ngrok http 7001
```

3. Once it's running, write down the web address that's tunneling to your localhost:7001 (ex: `http://55d5fex.ngrok.io`)

Configure and Run the Bot App

Configure and Run the Bot App

Download the project, swap in a few values, and start the bot up!

1. Download the node.js Webex Bot Starter template project with this command in your terminal:

```
git clone https://github.com/WebexSamples/webex-bot-starter && cd webex-bot-starter
```

2. Inside the project's root config-template.json file, use a code or text editor to swap out the following values:

```
{  
  "webhookUrl": "http://XXXXXXX.ngrok.io",  
  "token": "BOT_TOKEN_CREATED_FROM_EARLIER",  
  "port": 7001  
}
```

Configure and Run the Bot App

Download the project, swap in a few values, and start the bot up!

3. Note that when you turn off nGrok on your machine, the URL will change when you turn it back on, and you will need to update the config with the new address.
4. Save the changes and *rename* the file to **config.json**.

```
{
  "webhookUrl": "http://XXXXXXX.ngrok.io",
  "token": "BOT_TOKEN_CREATED_FROM_EARLIER",
  "port": 7001
}
```

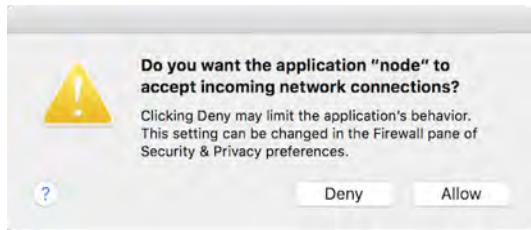

Configure and Run the Bot App

Download the project, swap in a few values, and start the bot up!

5. To start the bot application, run this command:

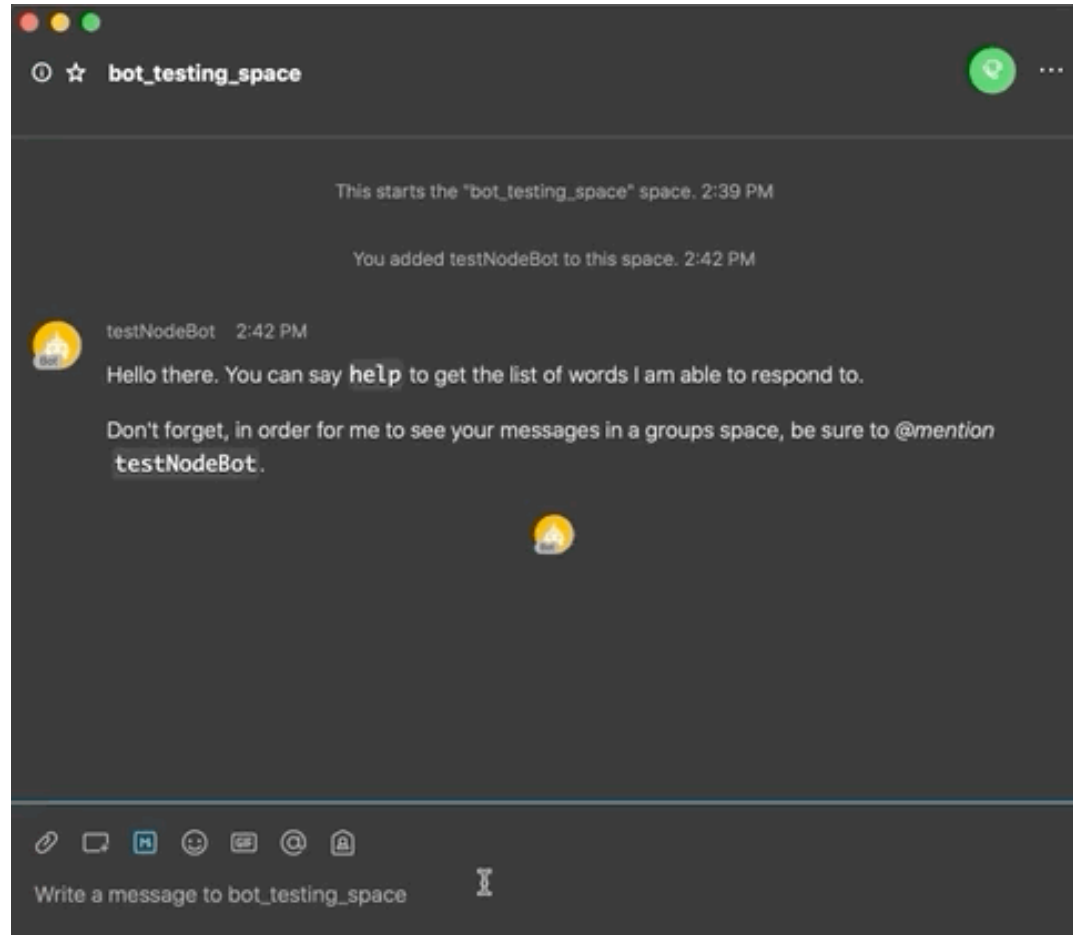
```
$ npm start
```

6. If you're using a Mac, you might receive a prompt asking about “node” accepting incoming network connections. If so, click Allow to load the server:



Interact with the Bot in Webex Teams and Inspect the Code

- Create a new group space in Webex Teams for testing your bot
- Follow the instructions it provides to explore some of the basic functionality of the bot



Going Further

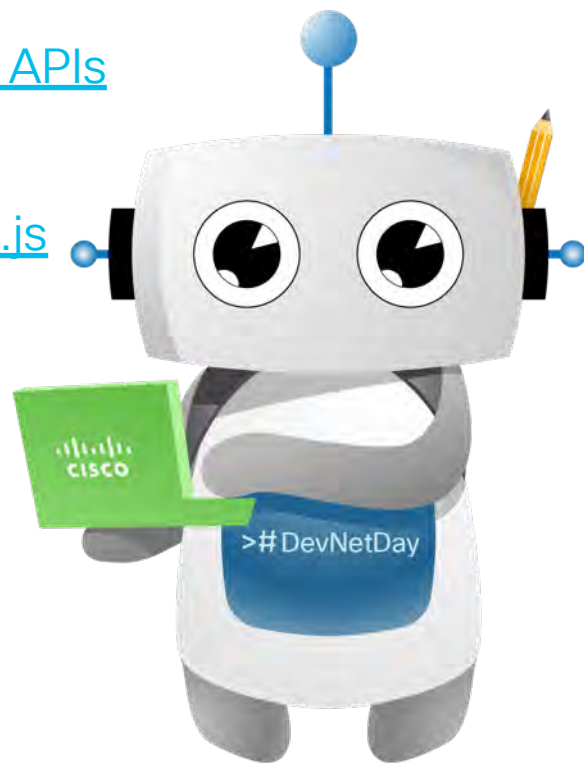
- To extend the application, open the `index.js` file in a text editor
- Inspect the code to see how the application handles user input and responds

```
52 //Process incoming messages
53
54 let responded = false;
55 /* On mention with command
56 ex User enters @botname help, the bot will write back in markdown
57 */
58 framework.hears(/help|what can i (do|say)|what (can|do) you do/i, function (bot) {
59   console.log('someone needs help! They asked ${trigger.text}');
60   responded = true;
61   bot.say(`Hello ${trigger.person.displayName}.`)
62     .then(() => sendHelp(bot))
63     .catch((e) => console.error(`Problem in help hander: ${e.message}`));
64 });
65
66 /* On mention with command
67 ex User enters @botname framework, the bot will write back in markdown
68 */
69 framework.hears('framework', function (bot) {
70   console.log("framework command received");
71   responded = true;
72   bot.say("markdown", "The primary purpose for the [webex-node-bot-framework]");
73 });
74
```


Helpful Resources

Explore More

- [DevNet Learning Labs: Getting Started with Webex APIs](#)
- [GitHub: Webex Node.js Bot Framework](#)
- [Blog: Dive Into the Webex Bot Framework for Node.js](#)
- Public# Webex-Bot-Node-Framework Support:
<https://eurl.io/#BJ7gmISeU>



Thank you



Possibilities

#CiscoLive | #DevNetDay