**Bot Demo**

**Creating the Bot:**

The Bot requires a single account in order to send and receive information to/from Spark. As Bot application’s are still in Beta at the time of this writing, you can use your own user account’s authorization token for testing. Otherwise, click the below link to add a Bot to your spark account. Once it has been created, you’ll need to save the authorization bearer token that is provided someplace safe. The token should never expire.

<https://developer.ciscospark.com/add-app.html>

To use your personal account token instead, simply click on your portrait in the top right to copy it. This code is valid for two years. <https://developer.ciscospark.com>

Note: Because this is a bot or bot-like demo, Oauth is not used in this process, so all authorization tokens referred to are at the account level and should already exist. They do not need to be manually created.

**Creating the Webhook:**

You’ll need a server setup capable of receiving HTTP POST requests. You can use a service like Ngrok to make your personal machine accessible to the world on a specific port for free.

<https://ngrok.com/>

Alternatively, Amazon Web Services offers a free EC2 micro instance for 1 year. Once you have an endpoint to use, create the webhook using the request on this page:

<https://developer.ciscospark.com/endpoint-webhooks-post.html>

If you are using a bot, make sure that the bearer token used is the bearer token of the bot.

Example Webhook:  
{

"id": "Y2lz111111112222222233333333",

"name": "BotDemo Project Room Awesome",

"targetUrl": "http://ec2-10-20-30-40.us-west-2.compute.amazonaws.com:10010",

"resource": "messages",

"event": "created",

"filter": "roomId=Y2lz12345678901234567890"

}

**Running the Code:**

The code included (bot\_demo.py) is a Python script that acts as a server running on port 10010. This port will need to be opened (or changed) on your server environment, in order for the webhook to reach the Python application. Messages that are posted in the room with the roomId that was used for the webhook are read by the application. The code is looking for slash commands. The following three in particular:

/batman

/batcave [Optionally any text following]

/batsignal

The bot will then reply to the room that one of the above commands is posted in.

In the provided code, you will need to edit the line near the bottom to include your bot or account’s bearer token:

bearer = "YOUR BEARER TOKEN HERE"

You will also need to make sure Python is installed on your system. For Mac and Linux systems, Python is installed by default. To run the server application, open a command terminal, and navigate to the folder that bot\_demo.py is in, then run:

python bot\_demo.py

**For the Future:**

Spark’s goal is to support @commands in the future (as opposed to a /command). That way, commands can be officially supported similar to mentions. It is recommended that for any bots underdevelopment, @commands are used instead of /commands.