

LABCOL-2293

Building Webex Teams bots using business logic templates

Vladimir Savostin

Sergey Sazhin

Introduction and Learning Objectives

Upon completion of this lab, you will get practical experience working with interface of Webex Teams bots, while hiding complexity and concentrating on business logic.

Doing this lab, you will build and run a Webex Teams bot for one of the business process types:

- Guest/campaign

- interview/test

- Cisco device insights

- TAC Self Service (errors, crashes)

During the lab you will:

* Learn bot’s capabilities in Webex Teams
* Use API calls to register the bot
* Integrate bots with database
* Make bot process photo and text
* Get familiar with business logic templates to use them in the future
* Learn the most usual errors with Webex Teams bots and how to handle them

During FIFA WorldCup 2018, guests were supported by the bots used as crucial part of social responsibility campaign.

In this lab, you will be presented with the platform, tools and components to build the bots in Webex Teams. You will also be shown how to use business logic templates to run different automation workloads on top of platform integration. You will also get recommendations regarding platform design decisions and how to solve the most usual problems. You will also learn how to run automation of business processes using bots interface.

Disclaimer

This training document is to familiarize with writing Webex Teams bots for social campaigns. Although the lab design and configuration examples could be used as a reference, it’s not a real design, thus not all recommended features are used, or enabled optimally. The code posted in this lab was tested in social campaign during FIFA WorldCup 2018 so it mighn’t work optimal and efficient in all environments. For the design related questions please contact your representative at Cisco, or a Cisco partner.

Table of contents

Introduction and Learning Objectives 2

Disclaimer 2

Table of contents 3

Scenario 4

Network Diagram 6

Option #1: Fix all issues without Webex Teams bot 7

**Step 1: Add yourself to the Webex Teams bot** 7

**Step 2: Answer questions to fix all issues in your network** 7

Option #2: Use Webex Teams bot to integrate with HelpDesk 8

**2.1 Write sample bot** 8

**2.2 Run internal Company’s event (Secret Santa)** 8

**2.3 Run external marketing campaign** 8

Option #3: Use Webex Teams bot to integrate with HelpDesk 10

**3.1 Write sample bot** 10

**3.2 TAC Self Service (errors, crashes)** 10

**3.3 Run interview/test** 10

Related Sessions at Ciscolive 11

Summary 11

Scenario

Disclaimer:

This is a work of fiction. Names, characters, businesses, places, events, locales, and incidents are either the products of the author’s imagination or used in a fictitious manner. Any resemblance to actual persons, living or dead, or actual events is purely coincidental. But it’s based on a true story.

**Imagine the following:**

There are many errors on the equipment in your network. Your department is understaffed and even despite of you’ve being working only with collaboration products for many years (congratulations, you are Collaboration SME!), your manager asks you to fix the issues by yourself. You know, your network is comprised of many different technologies:

* Routing & Switching
* Security
* Wi-Fi
* Collaboration

From one of the Cisco trainings, you remember there is ability in Cisco Webex Teams bots for the integration with external APIs to get information out of them. You know, there is Cisco API which allows you to find root causes of the issues searching the errors from equipment’s log.

**So, you have three options (see Figure 1):**

1. Option #1 – find root causes for all of the different technologies by yourself (tech level - basic).
2. Option #2 – write the Webex Teams bot, integrate it with Cisco API and give it to your support desk so they find and fix it for you (tech level - medium).
3. Option #3 – write the Webex Teams bot, integrate it with Cisco API and give it to your support desk so they find and fix it for you (tech level - advanced).

**Figure 1 – High-Level overview of the lab tasks**



Get started

Please connect to the lab with **Cisco AnyConnect** (you can get VPN server name and credentials from your lab ticket or from lab proctor):

* 1. Start **Cisco AnyConnect** on your laptop.
  2. Copy the **Host** URL from the **AnyConnect Credentials** and then paste it in the **URL Connection box** in the **AnyConnect** login window.
  3. Click **Connect**.  
     
  4. If you get a connection error, remove the “https://” part of the URL and try the connection again.
  5. Copy a User ID and the password from the **AnyConnect Credentials** and then paste each into the **Cisco AnyConnect** login window.
  6. Click **OK**.
  7. Click **Accept** on the window confirming your connection.
  8. When connected to your AnyConnect VPN session, the AnyConnect VPN icon is displayed in the system tray.
  9. To view connection details or to disconnect, click the AnyConnect VPN icon and then choose **Disconnect**.

See High-level Lab’s Topology below.

High-level Lab’s Topology



**Figure 2 ̶** **High-level Lab’s Topology**

Above is the lab’s toplogy. The central part of lab is Cisco Webex Teams bot, which is to be written on Python.

Below is high-level information about topology of the lab, please, read it carefully, before starting doing the lab.

Bots are similar to regular Webex Teams users. They can participate in 1-to-1 and group spaces and users can message them directly or add them to a group space. A special badge is added to a bot's avatar in the Webex Teams clients so users know they're interacting with a bot instead of a human.

A bot can only access messages sent to it directly. In group spaces, bots must be [@mentioned](https://collaborationhelp.cisco.com/article/en-us/p5k20o) to access the message. In 1-to-1 spaces, a bot has access to all messages from the user.

Bots do not, however, perform actions within Webex Teams on behalf of a Webex Teams user. If you're creating an application that needs to participate in Webex Teams and perform actions with a user's account, check out [Integrations](https://developer.webex.com/docs/integrations).

Bot’s integration with external entities

Depending on the path selected and chosen task, it will communicate with the following external entities:

1. Data Sources:

* PostgreSQL
* MongoDB
* Text files in JSON format

1. Gmail (via SMTP)

Bot’s integration with Webex Teams clients

In every task of the lab, a bot would be communicating with a Webex Teams client (in the course of this lab, web-based client of Webex Teams is used).

A bot can’t communicate directly with Webex Teams clients, but does it via Webex Teams Cloud.

Communication is established in the following way:

1. In order to deliver messages, bot sends HTTP POST to a client – directly through Webex Teams Cloud.
2. In order to get messages from a client – bot is programmed to listen to Webhook (see details below) from Webex Teams Cloud. Each time Webex Teams Cloud detect message from a client – it notifies a bot using Webhook (using HTTP POST).

JSON is used as format of sending data between Python bot and Webex Teams Cloud.

Flask

In order to be able to receive Webhooks from Webex Teams Cloud, a bot must be integrated with Web Server. In a course of the lab, [Flask](http://flask.pocoo.org/), an easy and powerfool framework would be used to run a web server on Python.

Ngrok

Since a bot is running on a server which isn’t reachable from the Internet, [Ngrok software](https://ngrok.com/) is used to host Public URLs using which bot’s web server could be reachable (for building webhook integration).

Notes:

* In the lab, client and server, while a logically separate entities in a drawing above, are physically hosted on the same machine.
* All script for bots are tested on Python 3.7.2 but should also work on another Python 3 releases.
* Cisco Webex Teams client exists in agent and agentless (web-based) form-factors. Throughout this lab web-based client would be used.

Option #1: Fix all issues without Webex Teams bot

Technical level of this task: **basic**

**1.1 Steps (Use TAC Self Service bot to find issue):**

* Add yourself to the existing Webex Teams bot.
* Get answers for the questions to fix all issues in your network (the bot, which is written in “Quiz” task in Option #2 is used here).
* Create and run sample bot (shows information about your Webex Teams account and echoes back received messages)

**Step 1: Add yourself to the Webex Teams bot**

Cisco Webex Teams client exists in agent and agentless (web-based) form-factors. Throughout this lab web-based client would be used.

1. Use the following link to open a web-based Webex Teams client:

[https://teams.webex.com/](https://teams.webex.com/spaces)

1. Enter email address[[1]](#footnote-1), press Next
2. Enter password1 and press **Sign In** button.
3. On the left corner in a Search tab, input name of Service Webex Teams Bot:

**CLEUR\_LABCOL2293\_SERVICE@webex.bot (see Figure 3)**



**Figure 3 ̶ Find a service bot (CLEUR\_LABCOL2293\_SERVICE@webex.bot)**

**Step 2: Answer questions to fix all issues in your network**

1. Webex Teams space would be created, so that you can chat with a bot.
2. Start putting bug ids into a chat box in a space with a bot (see **Figure 4**).

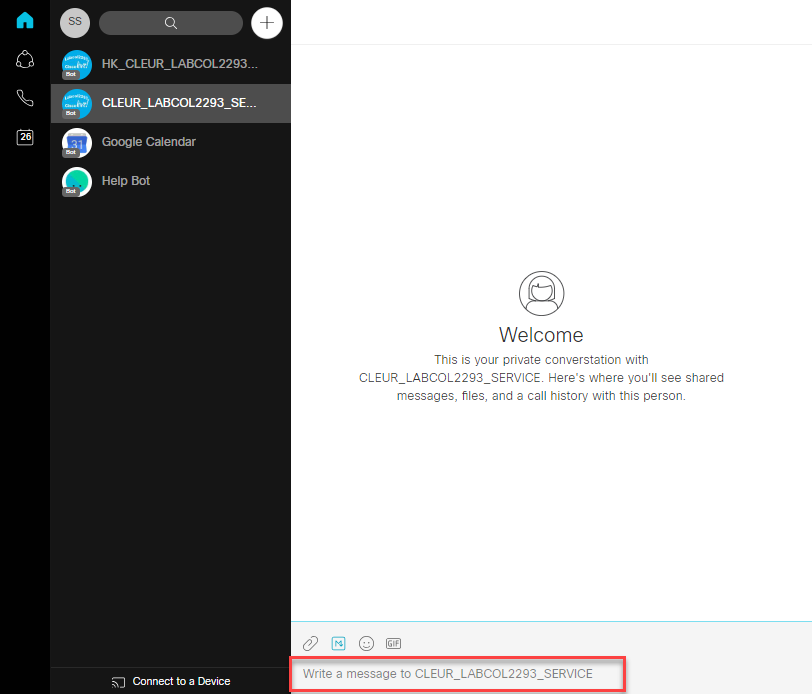
**/bug <bug\_id>**

You can user the following bug ids to test bot’s functionality:

/bug CSCdr72939

/bug CSCvb57414

/bug CSCuz44968



**Figure 4 ̶ Chat with a service bot**

**Steps 3 See how the bots are created:**

Create, start you own bot and interact with it in Webex Teams client.

All script for bots in this lab are written on Python 3[[2]](#footnote-2).

1. Create a bot on Webex Teams.
2. Create a [Webhook](https://developer.webex.com/docs/api/guides/webhooks) (so that a bot could listen to the messages of Webex Clients).
3. Copy code of sample bot from GitHub.
4. Replace Bot's Access Token in a configuration of a bot.
5. Start Ngrok.
6. Start Bot (Python script).
7. Open Webex Teams client and start chatting with a basic bot.

Create a bot on Webex Teams

There are two parts to create a bot:

* Create a “user” account on Webex Teams, using Webex for Developers.
* Create a program that provides the brains behind the created account.

1. Login to the following URL to create a new bot:

<https://developer.webex.com/login>

1. On the opened window:

Enter email address[[3]](#footnote-3)

Enter password3 and press **Sign In** button.

1. On the opened window:

Press **Start Building Apps** button

1. On the opened window:

Press **Create New App** button

1. On the opened window:

Press **Create a Bot** button

1. On the opened window fill in all the fields:

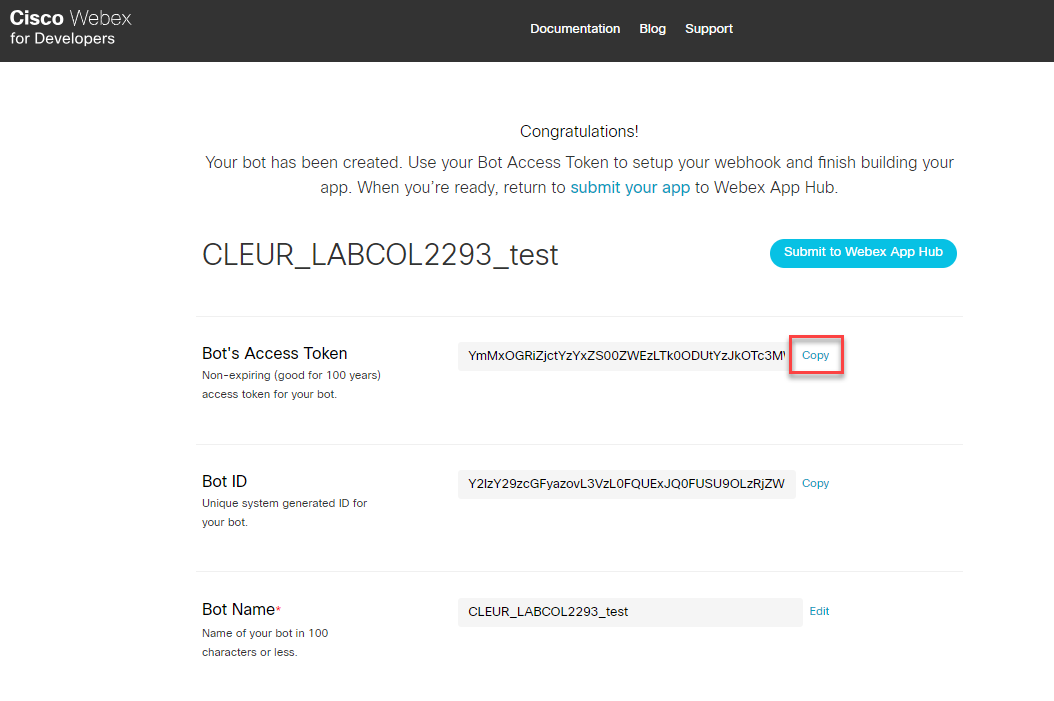
**Bot Name, Bot Username:** CLEUR\_LABCOL2293\_<random number>

**Icon:** Choose one of the default

**Description: CLEUR\_LABCOL2293 basic bot test**

1. Press **Add Bot** button
2. On the opened window:

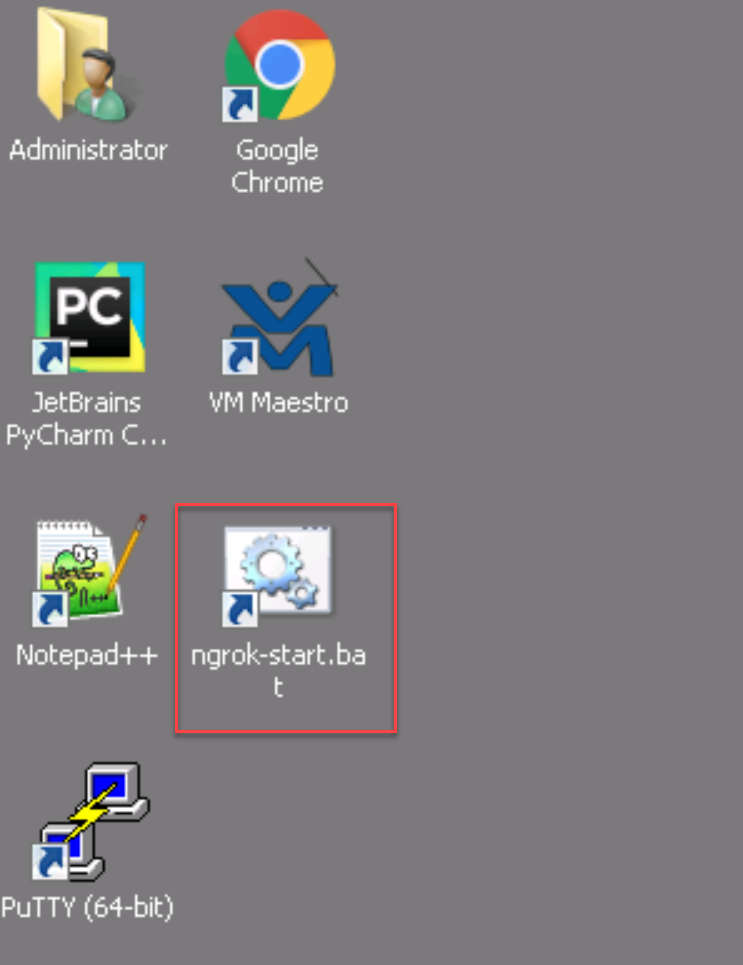
Copy **Bot’s Access Token** (see **Figure 5**), save it in a Windows notepad application.



**Figure 5 ̶ Copy bot’s access token**

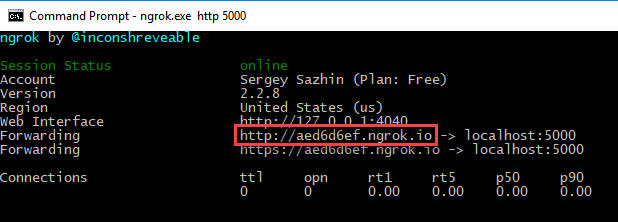
Start Ngrok

Click ngrok-**start.bat** on a Desktop (see **Figure 6**)



**Figure 6 ̶ Run ngrok-start.bat**

Windows command line would be opened, Ngrok will start to print out the data. Copy http “forwarding” URL (see **Figure 7**). All requests (in case of Webex Teams) coming to this URL would be forwarded to **localhost:5000** (Flask web server would be listening this socket). Copy this URL, which would be required to create a Webhook.



**Figure 7 ̶ Copy http “forwarding” URL**

Create a Webhook

In order to make a bot listen to the messages sent by Webex Clients, [Webhooks](https://developer.webex.com/docs/api/guides/webhooks) should be used:

<https://developer.webex.com/docs/api/v1/webhooks/create-a-webhook>

For each bot separate Webhook to be created. In another parts of the lab Webhooks are created automatically by script (based on forwarding URL created by Ngrok).

On the opened window fill in the fields in accordance with stated below, use Figure 8 for the guidance:

1. Authorization – **disable**
2. Bearer: <**Bot’s Access Token, copied in step 8 above>**
3. Name: **CLEUR\_LABCOL2293\_test\_bot**
4. TargetURL: <**“forwarding” URL from the previous step (Ngrok)>**
5. Resource: **messages**
6. Event: **created**

Press **Run** button

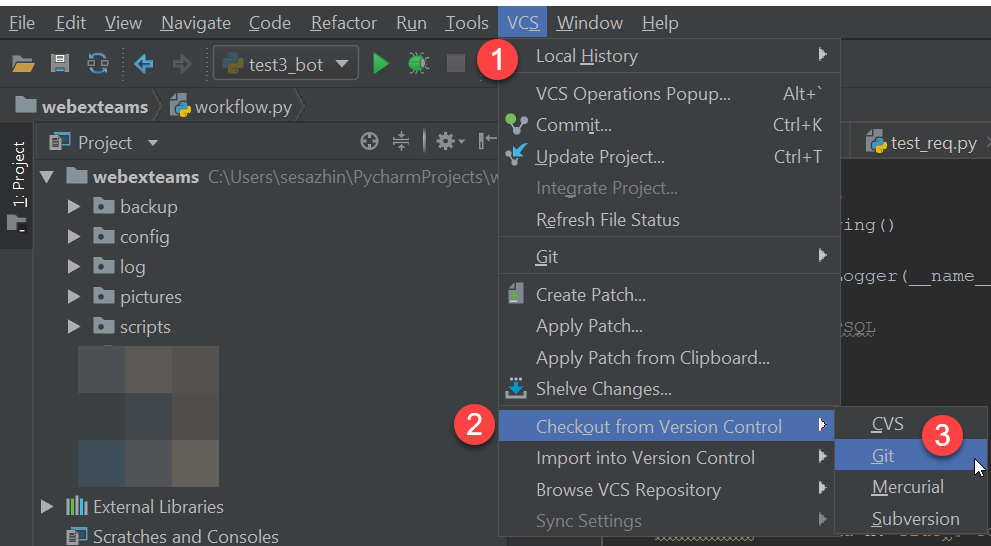


**Figure 8 ̶ Creating a webhook manually**

Copy code of sample bot from GitHub

Click “JetBrains PyCharm Community Edition” on a Desktop to run a PyCharm (IDE for Python)

Open: **VCS -> Checkout from Version Control -> Git**



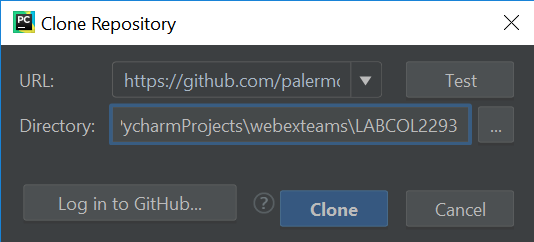
**Figure 9 ̶**

On the opened Windows use the following data to fill (see Figure 10):

**URL: <url on a GitHub>**

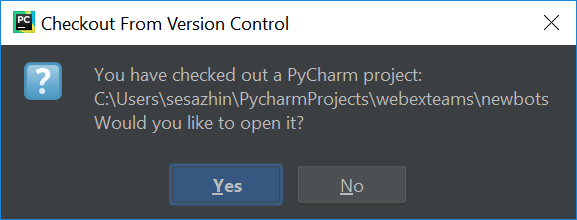
**Directory: <local directory on workstation>**

Press **Clone** button



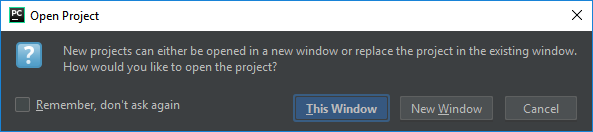
**Figure 10 ̶ Clone Repository Window**

On the opened Window press **Yes** to open the project checked out from GitHub:



**Figure 11 ̶ Chekout from Version Control Window**

On the opened Window press **This Window** to opened the project in the same window in PyCharm:



**Figure 12 ̶ Open Project Window**

Replace Bot's Access Token in a configuration of a bot

In a PyCharm, open a bot’s configuration file:

**SampleBot -> config -> config.yml**

In config.yml file, after **bot\_access\_token:** put <**Bot’s Access Token, copied in step 8 above>** (between quotes).

Start Bot (Python script)

On the left tab find a Python file with the code for sample bot:

**SampleBot -> server.py**

Open the file and press Shift + F10 to run the script

Open Webex Teams client and start chatting with a basic bot

1. Open a Web Browser with Webex Teams web-client, find the bot with created name: CLEUR\_LABCOL2293\_<random number>@webex.bot
2. In the created Webex Teams space, start chatting with a bot.

The following actions supported:

* **/whoami** – command that will instruct the bot to show information about your account in Webex Teams
* Input any phrase and bot will echo it back

Option #2: Use Webex Teams bot to integrate with HelpDesk

Technical level of this task: **medium**

**Option #2 Steps:**

2.1 Write sample bot.

2.2 Run internal Company’s event (Christmas Game).

2.3 Run external marketing campaign.

o that everyone

increase your vilibility inside the company, so that everyone knows you can use Webex Teams bots to solve current challenges in the innovative way.

You want to get rid of the job, related to the technologies you are not fun of, you want to write the bot to integrate it with Cisco API and give it to your support desk, so they find and fix all technical issues. But for now you think you should get more experience with Webex Teams bots before you start writing the bot for integration with Cisco external API.

To prove your colleagues you can write a Webex Team bot, you decided to show them a “quick win” and write a sample bot.

**2.1 Write sample bot**

Contains: creation of the sample bot which echoes messages from a user.

[Provide detailed steps to complete the task]

**2.2 Run internal Company’s event (Secret Santa)**

Everyone is very positive about your success with sample Webex Teams bot, nobody did it before you. Thanks, it’s Christmas time approaching, so, you know how to make another bot, which can help you to show you are well enough with bot’s writing. So, you come up to companie’s management with the following suggestion.

You will write a bot which will run the following Secret Santa game:

1. People willing to participate in Secret Santa game, join the Webex Teams space with bot.
2. After registration of all people has been finished (time ‘X’ has passed), the bot randomly choses pair for each participant.
3. The bot notifies participant about the pair. For this pair, the person must buy a gift and place it under a holiday tree.
4. Through the bot, participant can once ask a question from the person he has to buy a gift for (if he knows what to buy, he doesn’t need to this function, otherwise, he just clicks the button in a bot’s interface “ask”, after he clicks it, the button is just greyed out).
5. After the gift is bought, participant notifies the bot about it. If, one week before the Christmas, the bot didn’t receive confirmation from participant he bought a gift, the bot starts to notify participant once per day to buy a gift (until the confirmation is got).
6. Three days before the Christmas, the bot starts to notify a participant to place his gift under a holiday tree (once per day).
7. One day before the Christmas, bot notifies everyone to come up to a holidays tree and get gifts for them.

**2.3 Run external marketing campaign**

Your company is very proud of how the Secret Santa game were conducted. Everyone is inspired with truly innovative and modern way of running this task. So, you’ve got a query from marketing director to run external marketing campaign in a format of quest for a competition for a prize. To score points to win a prize, participants would need to answer bot’s questions about the brand of your company and post photos with themselves in a merchandise of your company (to get additional points).

**<You know there is existing business template for this task, so you don’t need to write the code from scratch on your own, just start using existing Python module with your Webex Teams bot. So, you accept this challenge>**

The bot must have the following logic:

1. People willing to participate, join the Webex Teams space with bot.
2. After registration of all people has been finished (time ‘X’ has passed), the bot randomly choses question out of a database and sends to each participant.
3. Participant answers the bot’s question and the bot either accept or decline an answer. If the bot accepts an answer – it adds points to the participant. If a participant sends photo with himself in a merchandise of your company, the bot resends this photo to marketing’s corporate email, so that this photo can be reviewed and if marketing specialist accepts it, he or she can accept this photo using a web link embedded into an email received from bot.
4. The bot sends the questions to a participant once per 3 days, it allows to post not more than 1 photo during the same period.
5. After campaign is finished, the bot generates table with results, determines the winner, and send this in a message the marketing specialist for approval. If it’s approved by marketing specialist, the bot sends congratulations to the winner.

Option #3: Use Webex Teams bot to integrate with HelpDesk

Technical level of this task: **advanced**

**Option #3 Steps:**

3.1 Write sample bot.

3.2 TAC Self Service (errors, crashes).

3.3 Run interview/test.

So, to get rid of the job, related to the technologies you are not fun of, you decided to tell the idea to your manager to write the bot to integrate it with Cisco API and give it to your support desk, so they find and fix all technical issues. Your manager is very positive about automation, but he isn’t sure you can make it. He knows you didn’t do this job before. To prove him you can do it, you decide to show him a “quick win” and write a sample bot.

**3.1 Write sample bot**

Contains: creation of the sample bot which echoes messages from a user.

[Provide detailed steps to complete the task]

**3.2 TAC Self Service (errors, crashes)**

This is your hour of triumph – after success in running external marketing campaign, your manager decided to let you invest your time in writing the bot to find root causes of the issues searching the errors from equipment’s log.

The following logic to be implemented by the bot:

1. Helpdesk worker posts the error from the log from network equipment
2. Bot connects to Cisco external API and searches for the similar issues
3. Bot returns possible bugs/description of the issues to a user
4. If a user accepts this as solution, the bot adds this information to the dashboard of the known issues (this dashboard could be seen by a management)
5. The bot notifies a helpdesk space in Webex Teams that this issue was identified.

**<You know there is existing business template for this task, so you don’t need to write the code from scratch on your own, just start using existing Python module with your Webex Teams bot>**

**3.3 Run interview/test**

Because you’ve solved many companies need using Webex Teams bots, your manager has decided to give you promotion. But you will need to interview new candidates for your former position. You are very busy, so don’t have enough time to talk to all of them to run preliminary checks for their technical skills. So, you decide to run the bot which will assess technical skills of candidates and decide whether you should talk in-person with them to assess further.

**<You know there is existing business template for this task, so you don’t need to write the code from scratch on your own, just start using existing Python module with your Webex Teams bot>**

*<Code to run questions one by one from the pool. Questions should be with pictures. In the end the script should show results. A student should understand this is the same test we were running in Option #1 😊>*

Related Sessions at Ciscolive

Extending the Collaboration Eco-System using Webex Teams APIs for Non-Developers - BRKCOL-2175

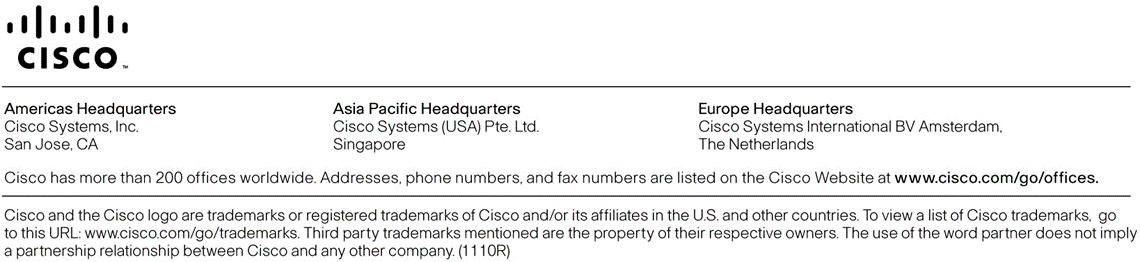
Cognitive Collaboration – AI and Machine Learning Assisted Collaboration - BRKCOL-1460

Summary

<Summary about what has been done>

1. List of Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Description** |
| SMTP | Simple Mail Transfer Protocol |
| IDE | Integrated Development Environment |
|  |  |
|  |  |
|  |  |



1. Use credentials from your lab ticket or from lab proctor [↑](#footnote-ref-1)
2. Python 3.7.2 release. [↑](#footnote-ref-2)
3. Use credentials from your lab ticket or from lab proctor [↑](#footnote-ref-3)