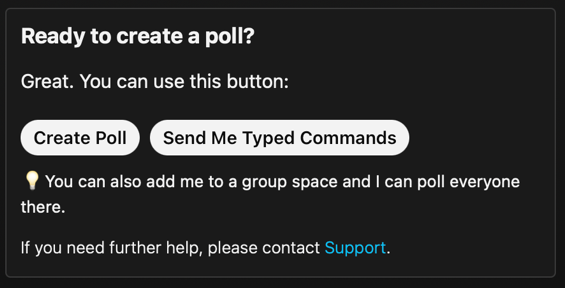
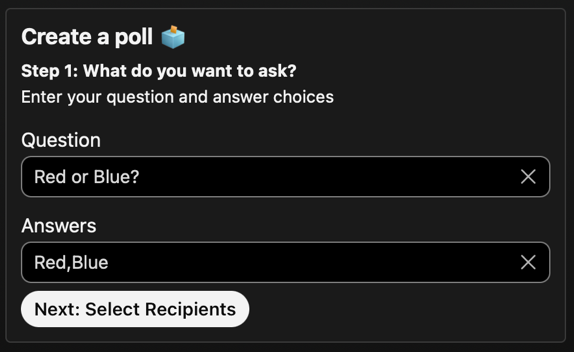
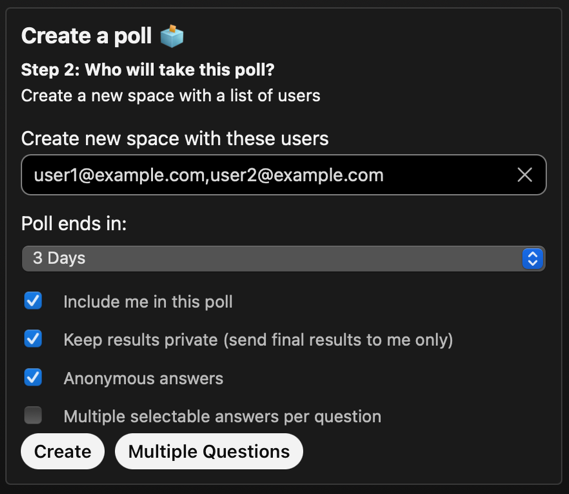
### Pollbot Open Source

We’re pleased to announce that Pollbot is now available as an open-source project that developers can modify to fit their own needs.

#### How it works

Pollbot is used for creating polls within Webex spaces. In its current design, if the bot is added to a group space then it will include every member of that space in any polls that are created in there. If the bot is used in a 1:1 space then you will be given the option to provide emails for any users you want to have included in the poll. Polls are created using [Buttons and Cards](https://developer.webex.com/docs/buttons-and-cards) that allow for easier user interaction than sending text commands, though that is possible as well.

Example of creating a poll in a 1:1 space:  
  
  


Example of creating a poll in a group space:  
Graphical user interface, text, application, chat or text message

Description automatically generated  
Graphical user interface, application

Description automatically generated

In order to use this code you will need to create your own [bot](https://developer.webex.com/docs/bots) and configure its access token in the .env file. The [example.env](https://github.com/WebexSamples/Pollbot-OpenSource/blob/56a6ba6a27b2b891943dc22c87cb16536259bbcb/example.env#L1) file has a list of required environment variables, both ones you will need to define as well as some universal ones. Once you’ve provided them in the file you can simply rename it to .env and the app will be able to read those variables when you start it.

The app requires three [webhooks](https://developer.webex.com/docs/webhooks) as well, be sure to use your bot token to create them.

1. Resource: messages, Event: created - pointed to the root of where you're hosting the app.   
   This webhook is triggered when someone sends the bot a message. The payload contains the messageId so the bot is able to [retrieve the message content](https://developer.webex.com/docs/api/v1/messages/get-message-details) in order to process the command.  
   Example: “targetUrl”:”<https://example.com>”
2. Resource: attachmentActions, Event: created - pointed to /cards.   
   This webhook is triggered when someone clicks an action button the cards, such as the “Create” button. The payload contains an attachmentActionsId so the data that was enter into the card and options that were select can be [retrieved](https://developer.webex.com/docs/api/v1/attachment-actions/get-attachment-action-details) and processed.   
   Example: “targetUrl”:”<https://example.com/cards>”
3. Resource: memberships, Event: all - pointed to /memberships.   
   This is used to track membership changes in a space, such as the bot being added/removed to/from spaces and if there is a moderation change in a space.  
   Example: “targetUrl”:”<https://example.com/memberships>”

This is an example of how the bot currently works when a user clicks the ‘Create Poll’ button on the initial card that’s sent, which triggers this [code](mailto:https://github.com/WebexSamples/Pollbot-OpenSource/blob/56a6ba6a27b2b891943dc22c87cb16536259bbcb/pollbot.py#L197):

else:  
 room\_type = room\_details.body.get("type")  
 card\_json = self.application.settings['card\_builder'].build\_question\_card(room\_id, room\_type, person\_id)

This will then call the [build\_question\_card](mailto:https://github.com/WebexSamples/Pollbot-OpenSource/blob/56a6ba6a27b2b891943dc22c87cb16536259bbcb/pollbot/src/card_builder.py#L197) function within /pollbot/src/card\_builder.py to build the actual card that will ask the user for their question and the possible answers they want to accept.

def build\_question\_card(self, room\_id, room\_type, person\_id):  
 card\_json = self.load\_card(Settings.create\_card)  
 if room\_type != "direct":  
 self.add\_options\_card(card\_json, person\_id)  
 card\_json["body"][4]["actions"][0]["data"].update({"person\_id":person\_id})  
 else:  
 container = card\_json["body"][1]["items"]  
 container[0]["text"] = "\*\*Step 1: What do you want to ask?\*\*"  
 container[1]["text"] = "Enter your question and answer choices"  
 actions = card\_json["body"][3]["actions"]  
 actions[0]["title"] = "Next: Select Recipients"  
 actions[0]["data"] = {"submit":"next", "person\_id":person\_id}  
 actions.pop(1)  
 return self.finalize\_card\_json(room\_id, "Adaptive Card - Create Poll", card\_json)

It pulls in the create\_card.json file from /pollbot/src/cards and then adds additional details to it that are specific to the user so it can be stored in the database and be tracked by the creators personId. After this it returns the updated card\_json variable and then [finalizes](mailto:https://github.com/WebexSamples/Pollbot-OpenSource/blob/56a6ba6a27b2b891943dc22c87cb16536259bbcb/pollbot/src/card_builder.py#L59) the JSON based on if the bot is being use in a group or 1:1/direct space.

def finalize\_card\_json(self, room\_id, markdown, card\_json, direct=False):  
 card = {  
 "markdown": markdown,  
 "attachments": [  
 {  
 "contentType": "application/vnd.microsoft.card.adaptive",  
 "content": card\_json  
 }  
 ]  
 }  
 if direct:  
 card.update({"toPersonId": room\_id})  
 else:  
 card.update({"roomId": room\_id})  
 return card

Then the card is posted to the space [here](mailto:https://github.com/WebexSamples/Pollbot-OpenSource/blob/941c0ea8696d5031329e621ddb69fbcde3de2a2d/pollbot.py#L296):

elif card\_json != {}:  
 yield self.application.settings['spark'].post\_with\_retries('https://webexapis.com/v1/messages', card\_json)

Which uses the [post\_with\_retries](mailto:https://github.com/WebexSamples/Pollbot-OpenSource/blob/56a6ba6a27b2b891943dc22c87cb16536259bbcb/pollbot/src/spark.py#L304) function in /pollbot/src/spark.py to handle any errors if they were to occur, such as being rate limited.

In addition to any code changes you’d like to do there are also card files that you can modify as well. In them there is a place to define a support link, [this is an example of one of them](https://github.com/WebexSamples/Pollbot-OpenSource/blob/56a6ba6a27b2b891943dc22c87cb16536259bbcb/pollbot/src/cards/help_active_card.json#L75), so your users can contact you should they run into any issues, though feel free to change them to fit your needs.   
  
To get started head over to the [Pollbot OpenSrouce](https://github.com/WebexSamples/Pollbot-OpenSource) repo on Github.