Webex Teams Hackathon 2018

Lab6 – Creating Webhook and REST API End Point for Webhook

Objectives

In this lab, you will complete the following objectives:

- Create a Webex Teams webhook
- Create a REST API POST endpoint for your Webhook
- Process the bot message by your back end

Background / Scenario

According to the background described in the previous lab we will create a Webex Teams webhook as a next step to handle actions in connection with our bot. Webhooks can be managed using Webex Teams APIs. You can get list of webhooks you created, create a new-, get the details of-, update- or delete an existing webhook. See https://developer.webex.com/docs/api/v1/webhooks for the details.

When completed, Webex Teams send notification to your back end each time when a message is sent to your bot. Your back end will get the message using Webex Teams API, process the message and send back an answer.

Required Resources

- Webex Teams client
- Postman application
- Python 3 with IDLE
- Python code files

Step 1: Create a webhook

In this step, you will create your first webhook.

a. Navigate to https://developer.webex.com/docs/api/v1/webhooks/create-a-webhook

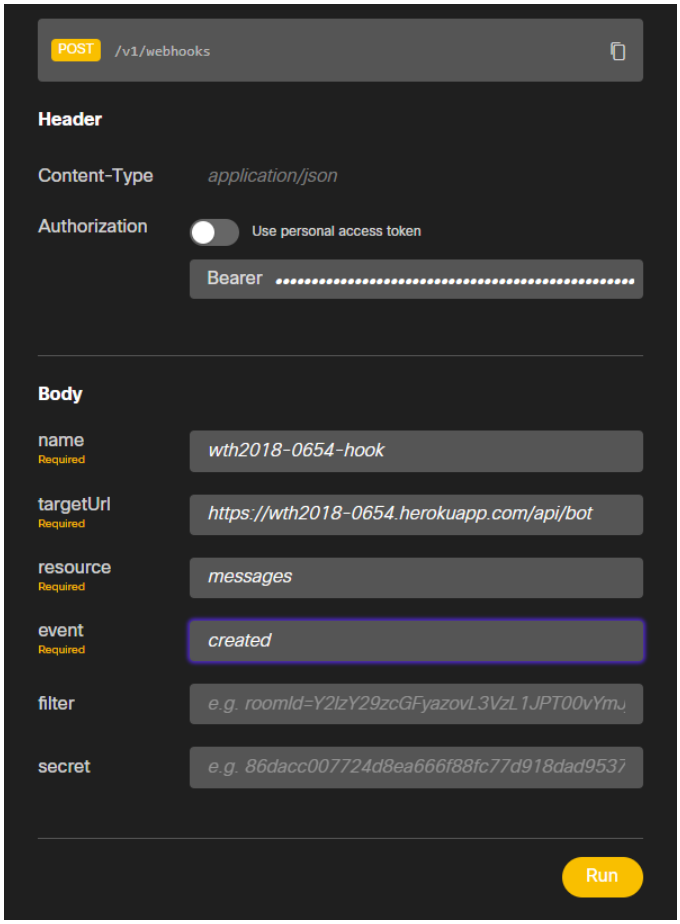
b. At the Authorization parameter switch off the “Use personal access token”

c. Add your bot’s access token after the Bearer text. If you do not find it, go to “My Webex Teams Apps” page, select your bot app, and regenerate the token.

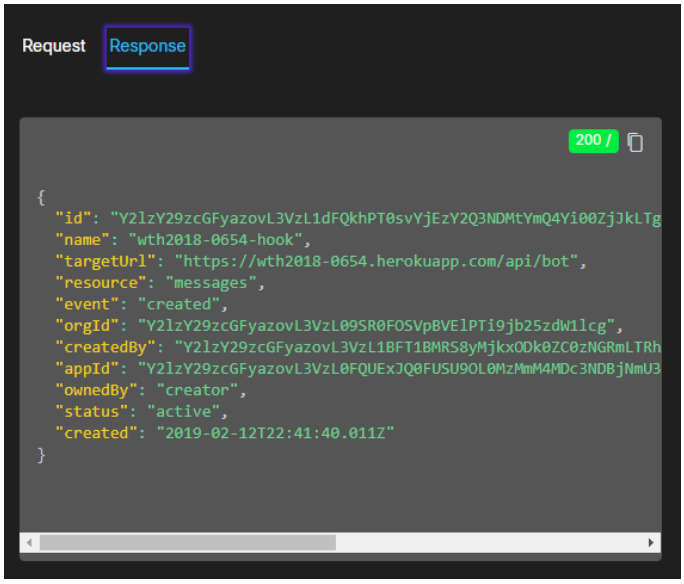
d. Under the body section give the following parameters:

name: `wth2018-<nnnn>-bot-hook`
targetUrl: `https://<your-heroku-host-name>/api/bot`
resource: messages
event: `created`
where `wth2018-<nnnn>` is your bot name and `https://<your-heroku-host-name>` is the URL where your back end can be accessed.

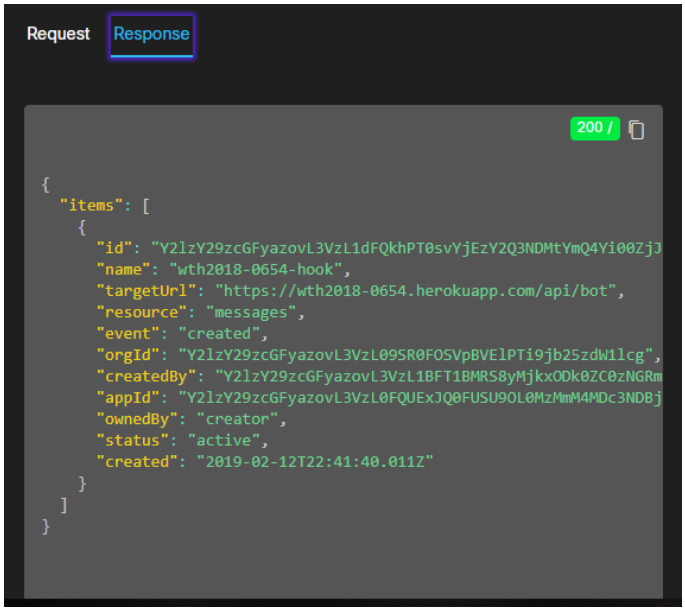
Leave the last two fields empty.



e. Click on “Run” button. You will get a similar response with the details of the new webhook.



f. List your existing webhooks. Click on List webhooks link at the left side bar. Use the same authorization setting than the previous step. You will get similar response. There is only one item in the items array which is the webhook you created in the previous step.



Step 2: Clone your back end project

a. Create a new directory and name it to `lab6-webhook`

b. Clone your `lab4-heroku` Github repo. In a command prompt at `lab6-webhook` directory type the following command:

```
git clone https://github.com/<your-github-username>/lab4-heroku.git .
```

Note: Do not miss dot at the end of the command.

c. Rename `lab4-heroku.py` file to `lab6-webhook.py`

d. Change `Procfile` content to `web: gunicorn lab6-webhook:app`

Step 3: Add new POST endpoint for your webhook

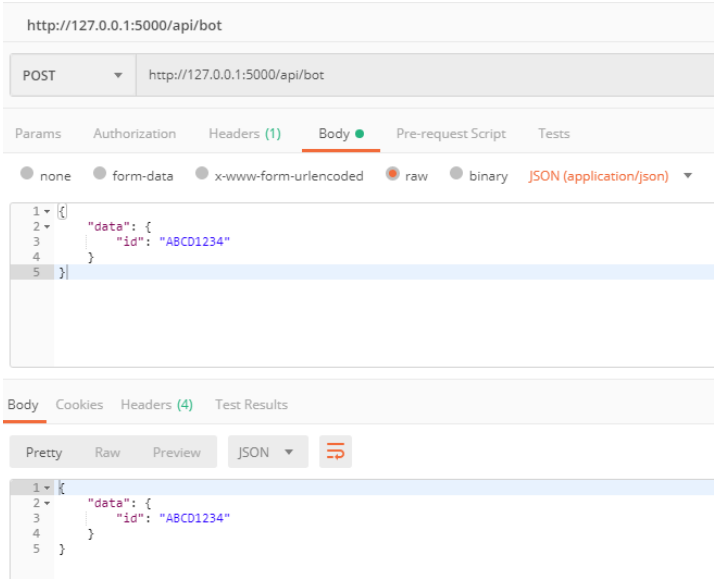
a. In your lab6-webhook.py file add a new endpoint:

```
@app.route("/api/bot", methods = ['POST'])
def hook():
	webhookMessage = request.json
	print(webhookMessage)
	messageId = webhookMessage["data"]["id"]
	print(messageId)
	return jsonify(webhookMessage)
```

b. Launch your back end app. In the command line type `python lab6-webhook.py`

c. Test the endpoint in Postman with the following JSON body:

```
{
	"data": {
		"id": "ABCD1234"
	}
}
```



d. Check the console output on the command line. You should see similar result.

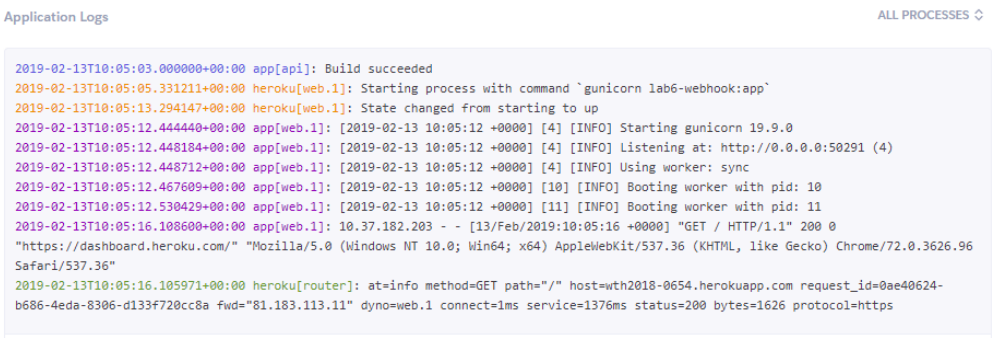
```
 * Serving Flask app "lab6-webhook" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
{'data': {'id': 'ABCD1234'}}
ABCD1234
127.0.0.1 - - [13/Feb/2019 10:41:43] "POST /api/bot HTTP/1.1" 200 -
```

e. Stop your back end app. Press `Crtl-C` in the command window.

f. Commit and push your changes.

```
git add .
git commit –m "/api/bot POST endpoint added"
git push
```

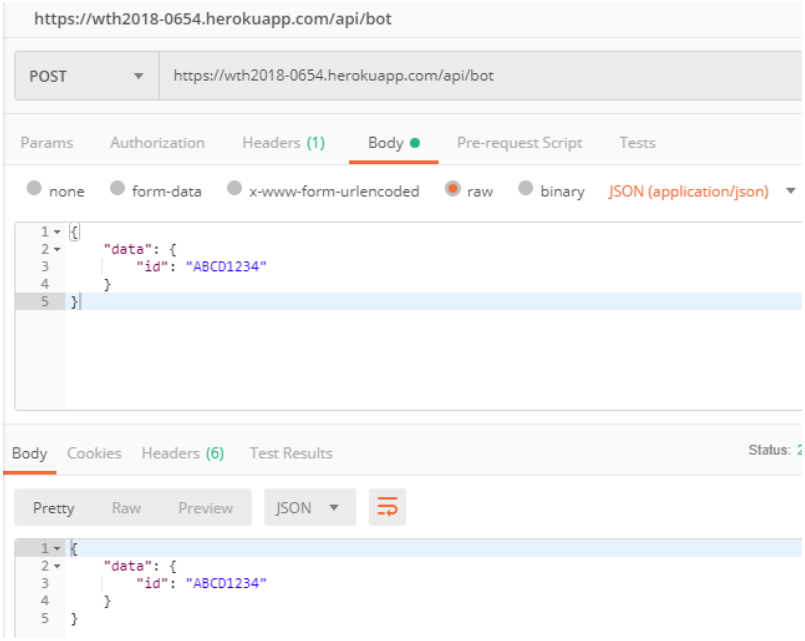
g. Login to Heroku (https://dashboard.heroku.com). Select your app and check the logs ("More" / "View logs") weather deployment was happened after your commit/push.



h. Open your app ("Open app")

i. Test the endpoint in Postman with your endpoint URL:

`https://<your-app-name>.herokuapp.com/api/bot`



j. See log massages:

```
2019-02-13T10:11:43.863289+00:00 app[web.1]: {'data': {'id': 'ABCD1234'}}
2019-02-13T10:11:43.864225+00:00 app[web.1]: ABCD1234
2019-02-13T10:11:43.868534+00:00 app[web.1]: 10.37.18.194 - - [13/Feb/2019:10:11:43 +0000] "POST /api/bot HTTP/1.1" 200 27 "-"
"PostmanRuntime/7.6.0"
```

Step 4: Send a message to your bot to check the webhook

a. Open your Webex Teams client. Select the “My demo space with my first bot” space then send the following message:

`@wth-<nnnn> Hello my bot!`

where `wth-<nnnn>` is your bot name.

b. Check the logs of your running back end on Heroku.

```
2019-02-13T10:23:04.835576+00:00 heroku[router]: at=info method=POST path="/api/bot" host=wth2018-0654.herokuapp.com request_id=94ee1ba2-3bf4-41a5-8424-7be100864393
fwd="18.221.216.175" dyno=web.1 connect=1ms service=8ms status=200 bytes=1300 protocol=https
2019-02-13T10:23:04.831568+00:00 app[web.1]: {'id': 'Y2lzY29zcGFyazovL3VzL1dFQkhPT0svYjEzY2Q3NDMtYmQ4Yi00ZjJkLTg4NzMtMTA0ODYyYTI0ZDg5', 'name': 'wth2018-0654-hook', 'targetUrl':
'https://wth2018-0654.herokuapp.com/api/bot', 'resource': 'messages', 'event': 'created', 'orgId': 'Y2lzY29zcGFyazovL3VzL09SR0FOSVpBVE1lPTi9jb25zdWl1cg', 'createdBy':
'Y2lzY29zcGFyazovL3VzL1BFT1BMRS8yMjkxODk0ZC0zNGRmLTRhM2EtYjRjMy04YjQ5ZGZkYzE4ZDU', 'appId':
'Y2lzY29zcGFyazovL3VzL0FQUExJQ0FUSU9OL0MzMmM4MDc3NDBjNmU3ZGYxMWRhZjE2ZjIyOGRmNjI4YmJjYTQ5YmE1MmZlY2JiMmM3ZDUxNWNiNGEwY2M5MWFh', 'ownedBy': 'creator', 'status': 'active',
'created': '2019-02-12T22:41:40.011Z', 'actorId': 'Y2lzY29zcGFyazovL3VzL1BFT1BMRS84NWQ5NTg3My1jNTZlLTRhODgtOWIwOC0wMjU5NTIxNWFkZmE', 'data': {'id':
'Y2lzY29zcGFyazovL3VzL01FU1NBR0UvNTkwM2E0NzAtMmY3OS0xMWU5LWIyYzYtYWI5ODIzZjA3Mzdm', 'roomId': 'Y2lzY29zcGFyazovL3VzL1JPT00vODBlY2NjMDAtMmVlZC0xMWU5LWFiYmUtZD1jZTc2MGI0ZDA2',
'roomType': 'group', 'personId': 'Y2lzY29zcGFyazovL3VzL1BFT1BMRS84NWQ5NTg3My1jNTZlLTRhODgtOWIwOC0wMjU5NTIxNWFkZmE', 'personEmail': 'wth@http-foundation.hu', 'mentionedPeople':
['Y2lzY29zcGFyazovL3VzL1BFT1BMRS8yMjkxODk0ZC0zNGRmLTRhM2EtYjRjMy04YjQ5ZGZkYzE4ZDU'], 'created': '2019-02-13T10:23:04.247Z'}}
2019-02-13T10:23:04.831657+00:00 app[web.1]: Y2lzY29zcGFyazovL3VzL01FU1NBR0UvNTkwM2E0NzAtMmY3OS0xMWU5LWIyYzYtYWI5ODIzZjA3Mzdm
2019-02-13T10:23:04.837879+00:00 app[web.1]: 10.9.210.147 - - [13/Feb/2019:10:23:04 +0000] "POST /api/bot HTTP/1.1" 200 1146 "-" "Squared Scheduler/1.0"
```

In the second last line you can see the id of the message which is the output of `print(messageId)` command of your app.

Step 5: Get the message using Webex Teams API

a. Install `requests` package if you haven’t done so before. In the command line at your project directory type the following command:

```
pip install requests
```

b. Add `import requests` command to the beginning of your project file.

c. Update `requirements.txt` by issuing the following command at the command prompt:

```
pip freeze > requirements.txt
```

d. Get the message part using Webex Teams message endpoint (https://api.ciscospark.com/v1/messages/<messageId>)

Add the following commands before your `/api/bot` endpoint before the return statement.

```
      url = "https://api.ciscospark.com/v1/messages/" + messageId
      r = requests.get(url, headers={'Authorization': 'Bearer <your-bot-
access-token> '})
      message = r.json()["text"]
      print(message)
```

where `<your-bot-access-token>` is your bot access token.

k. Commit and push your changes.

```
git add .
git commit –m "get bot message"
git push
```

e. After the deployment process finished send a new message to your bot: `@wth2018-<nnnn> Hello my bot`

f. Check the logs.

In the second last line you can see the message which is the output of `print(message)` command of your app.

Step 5: Reply for the message by the name of your bot

g. Send reply if your bot mentioned in the message. Add the following lines into your /api/bot endpoint after print(message) command.

```
      mentionedPeopleId = webhookMessage["data"]["mentionedPeople"][0]
print(mentionedPeopleId)
      if mentionedPeopleId ==
"Y2lzY29zcGFyazovL3VzL1BFT1BMRS8yMjkxODk0ZC0zNGRmLTRhM2EtYjRjMy04YjQ5ZGZkYzE4
ZDU":
            roomId = r.json()["roomId"]
            url = "https://api.ciscospark.com/v1/messages"
            r = requests.post(url, headers={'Authorization': 'Bearer
YTVlODc5MTgtZjBkYy00ZDVlLTliNGItZjUyYmY4YTM4NWQyNDY0YTNiYTQtMzIz_PF84_consume
r'}, data={'roomId': roomId, 'text': 'Hello from your bot!'})
```

Change mentionedPeopleId value to your bot id.

h. Send a new message to your bot: `@wth2018-<nnnn> Hello my bot again!`