

Contents

1	Add a webhook for your Slack workspace	1
2	Run the bot locally	1
3	Schedule the program	2
4	Conclusion	2

— title: "Have a bot haunt your Slack workspace" date: 2020-06-05
categories: ["Programming", "Web"] tags: ["python", "web"]
type: "post" —

In this article we'll see how to code your simple bot for Slack, and schedule it to run regularly. As some testing might be necessary, I recommend to create a dummy Slack workspace for this tutorial.

1 Add a webhook for your Slack workspace

We first need to configure a web hook to communicate to our Slack workspace. Go to this url, choose your channel and click on add a webhook. In this tutorial, I choose to post to the `#random` channel. The rest of the configuration can left as is, so you can save it.

Copy your web hook url which should have the form `https://hooks.slack.com/services/SOME_GIBBERISH_CODE`.

2 Run the bot locally

Let's test locally the connection to the webhook we just created. We'll use python library `slackweb`. The code is straightforward. We'll put this in `slackbot.py`.

```
#!/bin/python3
#-*- encoding: utf-8 -*-
import slackweb
HOOK_URL='https://hooks.slack.com/services/SOME_GIBBERISH_CODE.'
slack = slackweb.Slack(url=HOOK_URL)
slack.notify(text="Hello world!")
```

Replace your the content of `HOOK_URL` by your hook url copied from the previous step. Then run the code and check if it works; you should see in `#random` a message from your program. The first line of the above code

specifies the location of the python interpreter to run, and the second line should be used if the file contains unicode characters, such as japanese kanji or french é, à, etc. {{< figure src="slackbot-hello_{world}.jpg" width="auto" caption="" >}}

3 Schedule the program

You could decide to schedule the program to run on your own computer, but if you want it to run even when your computer is shut down, we need a web server. PythonAnywhere offers free server space and python environment for development purpose. So let's create an account and head to our PythonAnywhere dashboard. {{< figure src="pythonanywhere_{dashboard}.jpg" width="auto" caption="" >}} Go to the *Files* tab and upload your file `slackbot.py` to your directory `/home/your_username/`. {{< figure src="pythonanywhere_{files}.jpg" width="auto" caption="" >}} Then, let's install the dependancy. Open a bash terminal and run `pip3 install --user slackweb`. {{< figure src="pythonanywhere_{bash}.jpg" width="auto" caption="" >}} Inside the same terminal, we can check that everything works. Run `python3 slackbot.py`, and see if the message appears again in `#random` channel.

Scheduling is done in the *Task* tab. Create your task by specifying a frequency and the command to run: `/home/adamoudad/slackbot.py`. You can see the expiry date of your task when you created it, which is probably roughly one month. For the free plan, PythonAnywhere requests you to log at least once a month for your scheduled tasks to remain valid. {{< figure src="pythonanywhere_{tasks}.jpg" width="auto" caption="" >}}

4 Conclusion

That's it! you have your Slack bot up and running on the web. And it will haunt your Slack workspace until the end of times... cordially.