

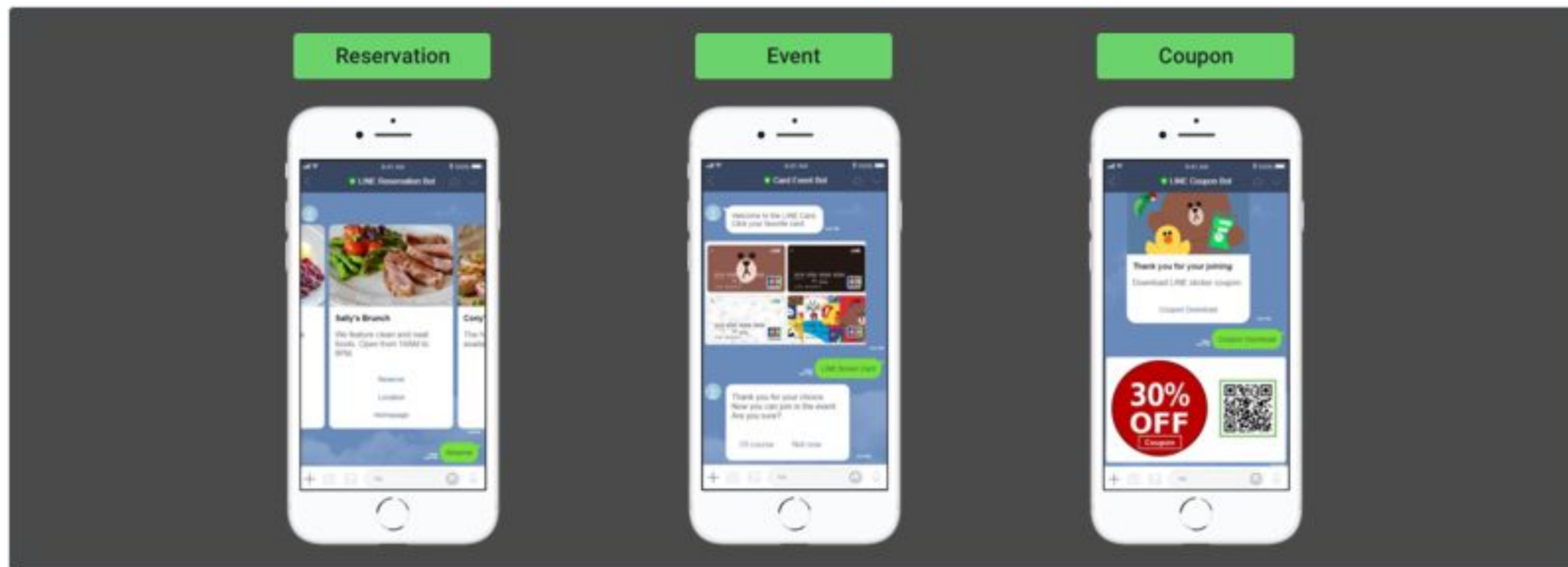
計算理論 110-1 期末專題

FSM Line Bot, TOC 110-1

TA: 張財實, 郭哲瑋, 孫毅夫, 陳冠廷



Build a Line Bot with FSM



Test your app (channel)

LINE Developers

ProductsDocumentsAPI referenceNewsFAQCommunityBlog

Search

Notifications

Profile

Console home

Providers

Search...

Admin

Notifier

TOC-TA

計理助教

Tools


Support

TOP > TOC-TA > NCKU-TOC-TA > Messaging API

Bot basic ID

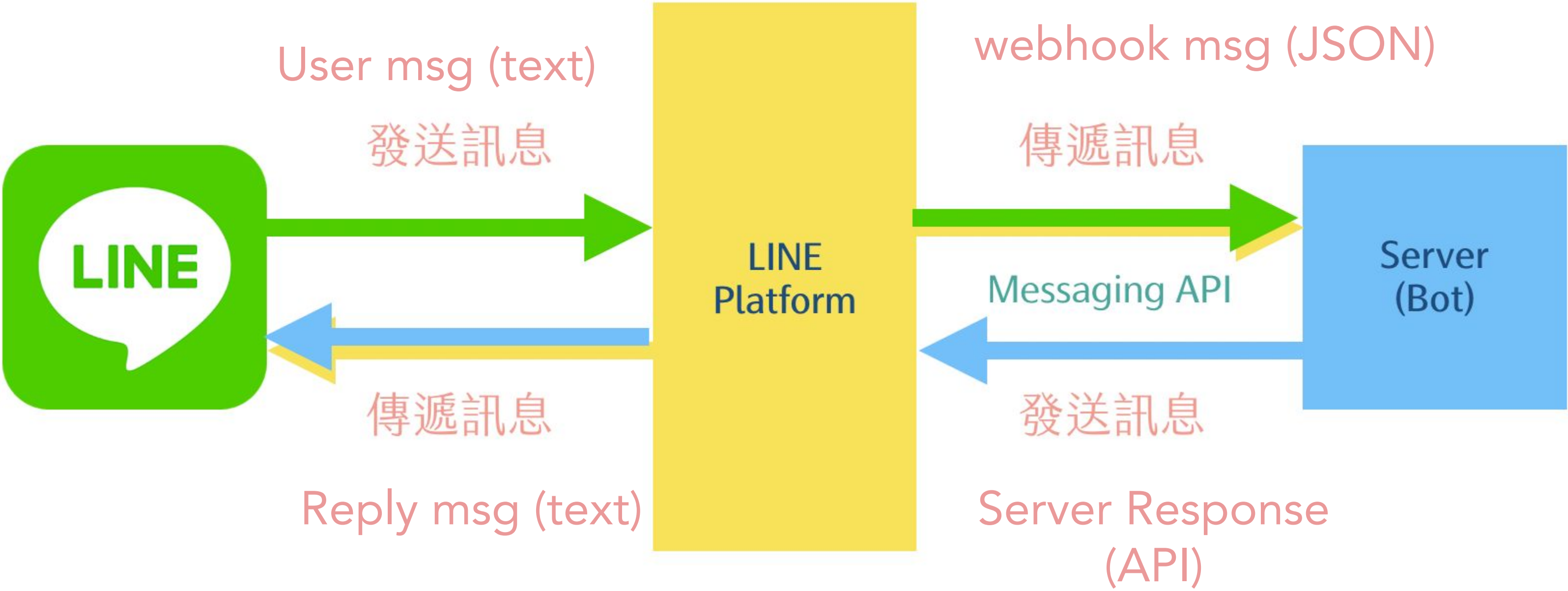
@738ulcbp

QR code



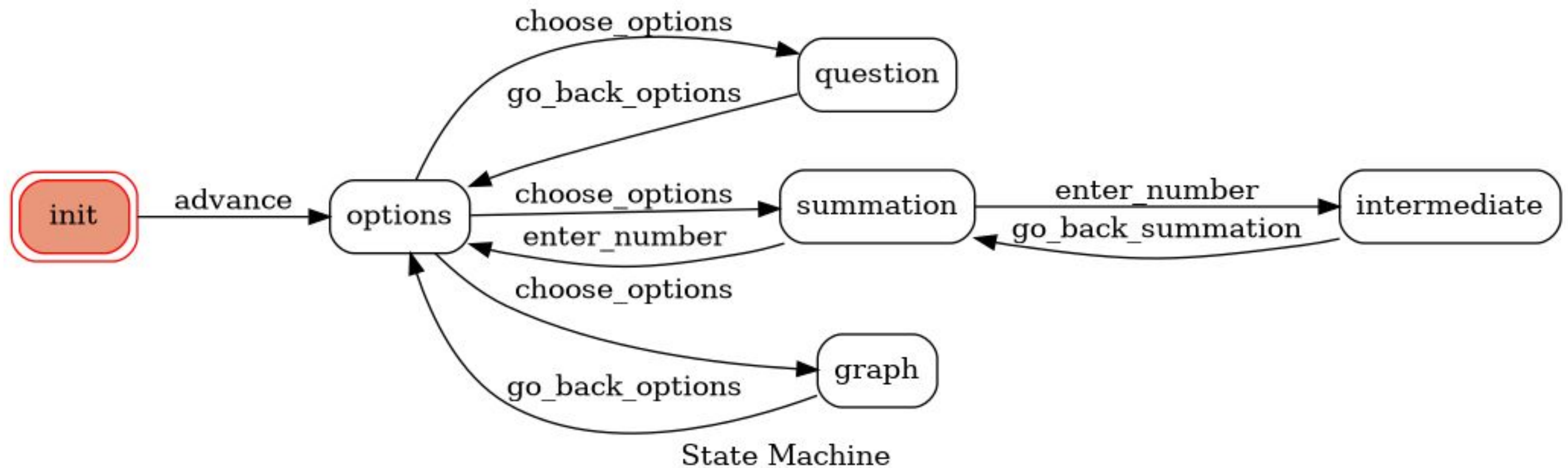
Scan this QR code with LINE to add your LINE Official Account as a friend. You can share the code with others.

Below shows API call underneath the system. Main work of this project is to build the Server (bot).



Use finite-state-machine logics to construct the bot.

- In different state, user should receive different reply.
- Line Bot should be stateful, not stateless for this project.



Previous Works

- [Line Fitness](#) - 黃政瑋
- [Daily-NBA](#) - 周鼎軒
- [Currency Related](#) - 葉家彥

- Project timeline
- Project roadmap
- Sample code
- Grading
- Documentation and Tutorial
- Reference
- Precautions

1. Project announcement (2021-12-08)
2. Fill in Github repo for grading (Google Forms)
3. Deadline (2022-01-02)
 - Grading is based on the latest commit before deadline
4. Demo reservation (Google Forms)
5. On site demo

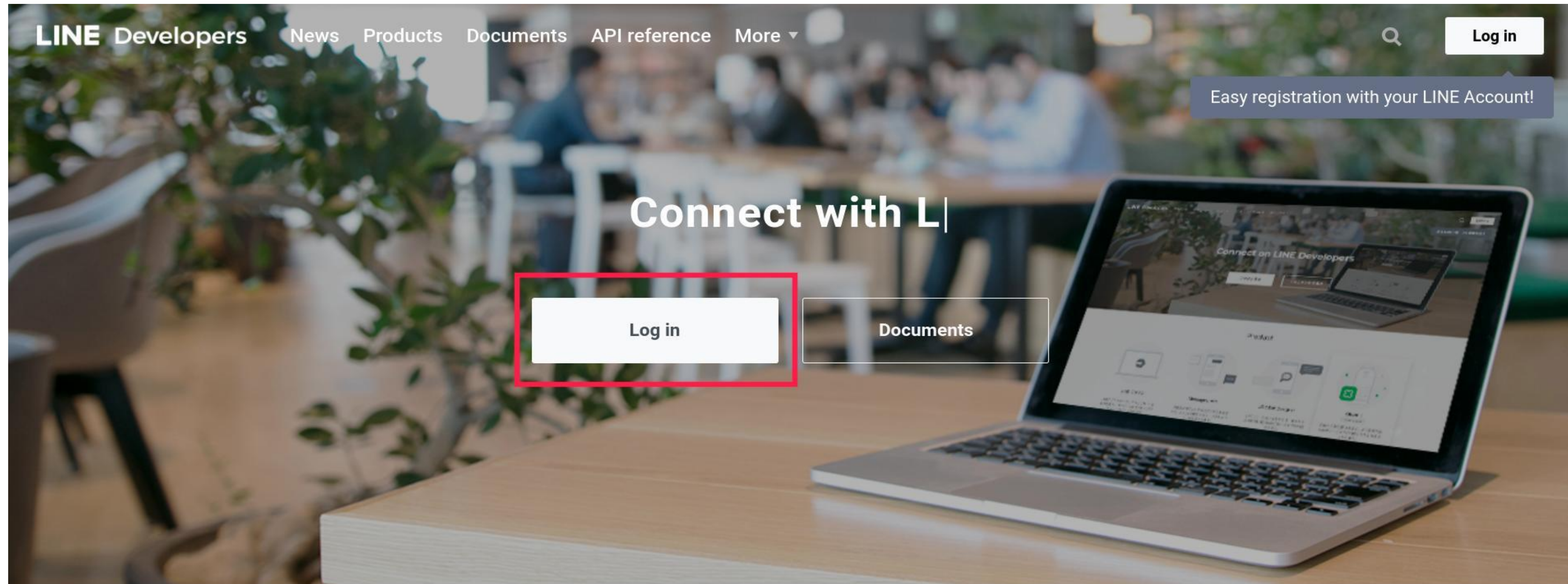
1. Have Python installed
2. Become a LINE developer
 - Create LINE Bot channel
 - Obtain access token and key
3. Build a simple server (Bot server)
4. Binding LINE bot
5. Test and develop
 - Read read read the docs!
6. Deploy
7. Show your work

[Python installation guide](#)

Suggestion:

- Use pipenv or virtualenv to create your python environment

[Register and Login!](#)



Products

LINE account

LINE Business ID

使用LINE帳號登入

或

使用商用帳號登入



建立帳號

使用LINE商用ID須遵從服務條款等規定，登入後即視為您同意相關條款內容。


[? 關於LINE商用ID](#)

繁體中文 ▾

[Help](#) [服務條款](#) © LINE Corporation

LINE Developers [Products](#) [Documents](#) [API reference](#) [News](#) [FAQ](#) [Community](#) [Blog](#)  

Console home

Providers 

Admin

Notifier


TOC-TA

Tools

Support

TOP

Providers (2) **Create**





Provider	Role
Notifier	Admin
TOC-TA	Admin


© LINE Corporation [Terms and policies](#) [About trademarks](#) [Need support? Please use our inquiry form](#)

Family sites

English

LINE Developers [Products](#) [Documents](#) [API reference](#) [News](#) [FAQ](#) [Community](#) [Blog](#)  

Console home

Providers 

Admin


Notifier

TOC-TA

Tools


Support

TOP

Providers (2) 

Create a new provider

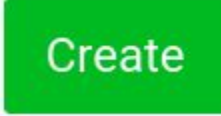
Your desired name

Provider name 

✓ Don't leave this empty

✓ Don't use special characters (4-byte Unicode)

✓ Enter no more than 100 characters



© LINE Corporation [Terms and policies](#) [About trademarks](#) [Need support? Please use our inquiry form](#)

Family sites

English

LINE Developers[Products](#)[Documents](#)[API reference](#)[News](#)[FAQ](#)[Community](#)[Blog](#)

Console home

Providers

Search...

Admin

Notifier

TOC-TA

計理助教

Tools

Support

TOP > 計理助教

This provider doesn't have any channels yet

To create one, choose a channel type below



Create a LINE Login channel



Create a Messaging API channel

© LINE Corporation[Terms and policies](#)[About trademarks](#)[Need support? Please use our inquiry form](#)

Family sites

English

LINE Developers

[Products](#)[Documents](#)[API reference](#)[News](#)[FAQ](#)[Community](#)[Blog](#)

Search

Console home

Providers

Search...

Admin

Notifier

TOC-TA

計理助教

Tools

Support

TOP

Create a channel

Fill in the blank

Channel type

Messaging API

✓ Don't leave this empty

Provider

計理助教

✓ Don't leave this empty

Channel icon

optional

© LINE Corporation

[Terms and policies](#)

[About trademarks](#)

[Need support? Please use our inquiry form](#)

Family sites

English

After channel creation, on Basic Setting, issue channel secret

LINE Developers

ProductsDocumentsAPI referenceNewsFAQCommunityBlog

Search

Profile

Console home

Providers

Search...

Admin

Notifier

TOC-TA

計理助教

Tools

Support

TOP > 計理助教 > Testing

optionalEdit

App typesBot

PermissionsPROFILE

Channel secretIssue

Assertion Signing KeyIssue

© LINE CorporationTerms and policiesAbout trademarksNeed support? Please use our inquiry form

Family sitesEnglish

On messaging API settings, at the button, gather channel access token

The screenshot shows the LINE Developers console interface. The top navigation bar includes links for Products, Documents, API reference, News, FAQ, Community, and Blog. The left sidebar contains links for Console home, Providers (highlighted in green), Admin, Notifier, TOC-TA, 計理助教, Tools, and Support. The main content area displays the Messaging API settings for the 計理助教 channel. It includes sections for Auto-reply messages (Enabled) and Greeting messages (Enabled), each with an Edit button. The 'Channel access token' section is highlighted with a red box and contains the text 'Channel access token (long-lived)' and an 'Issue' button. The footer includes copyright information for LINE Corporation and links to Terms and policies, About trademarks, and an inquiry form. There are also dropdown menus for Family sites and English.

LINE Developers Products Documents API reference News FAQ Community Blog

Console home

Providers

Search...

Admin

Notifier

TOC-TA

計理助教

Tools

Support

TOP > 計理助教 > Testing > Messaging API

Auto-reply messages Enabled [Edit](#)

Greeting messages Enabled [Edit](#)

Channel access token

Channel access token (long-lived)

[Issue](#)

© LINE Corporation Terms and policies About trademarks Need support? Please use our [inquiry form](#)

Family sites English

▶ Building the Bot Server

19

Building a HTTP Server

- GET
- POST
- PUT
- DELETE

Choose a framework (Python)

- Flask
- Django
- Fastapi

Use FSM tools (pip install transitions) to build response logic

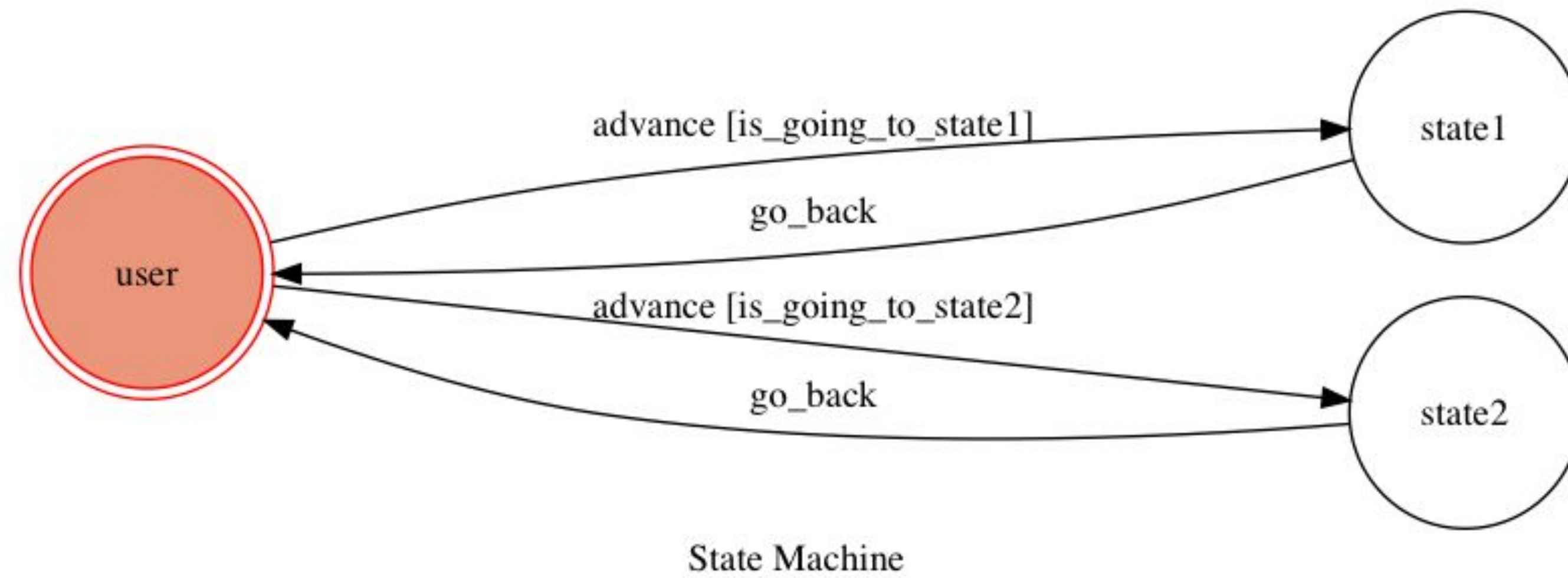
Sample Code



<https://github.com/NCKU-CCS/TOC-Project-2020>

Building the Bot Server - FSM Graph

FSM graph for sample code.



Writing a HTTP server - Overview

img
.env.sample
.gitignore
Aptfile
Pipfile
Pipfile.lock
Procfile
README.md
app.py
fsm.png
fsm.py
requirements.txt
utils.py

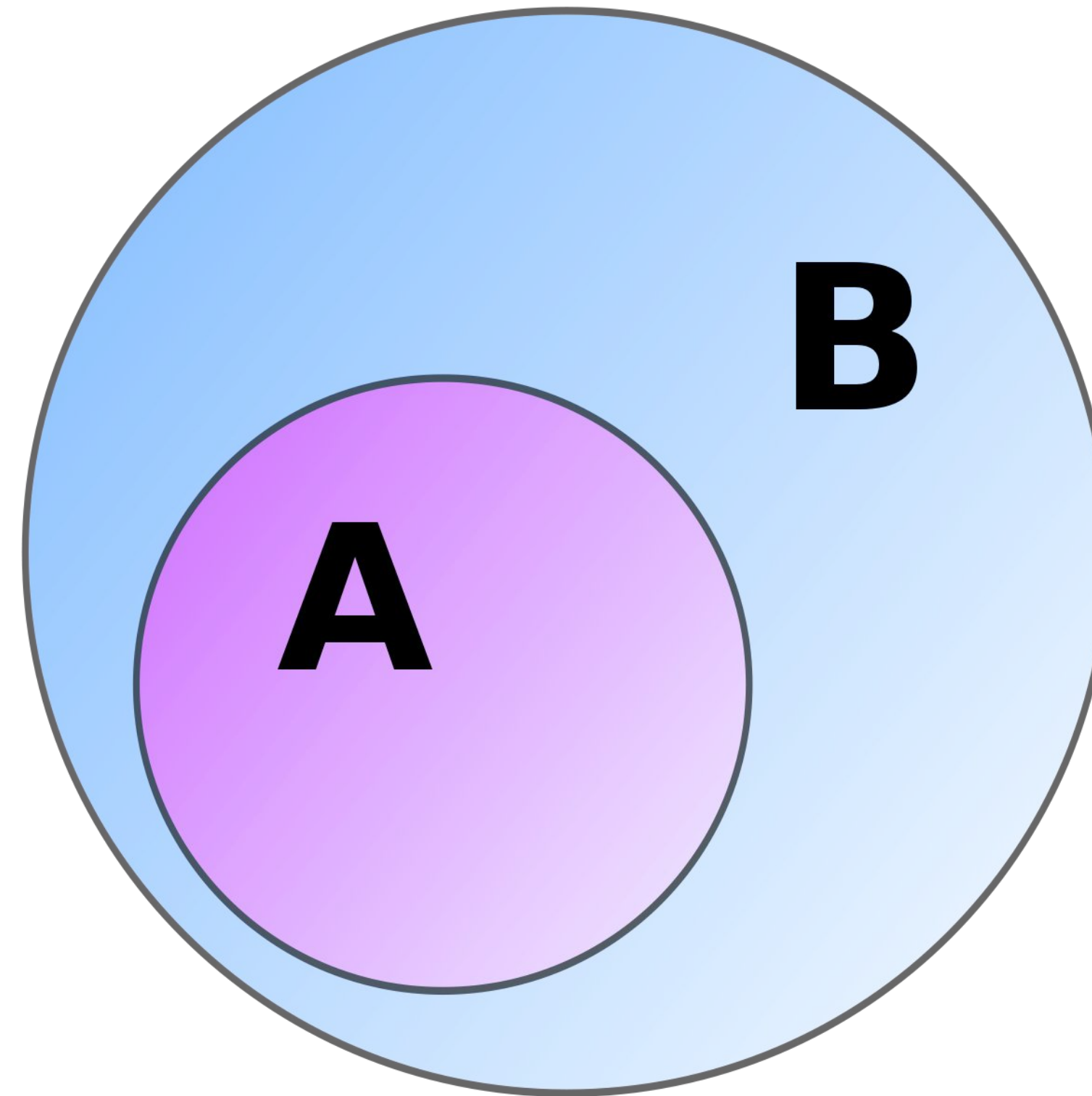
Flask server overview

- app.py - main file
- fsm.py - file handles fsm logic
- .env - file stores environment variable
- utils.py - helper function

Writing a HTTP server - Python Environment

Best practice for writing python project is to develop your code under an enclosed environment.

- virtualenv
- poetry
- pipenv - [tutorial](#)



B - denoted as system python environment

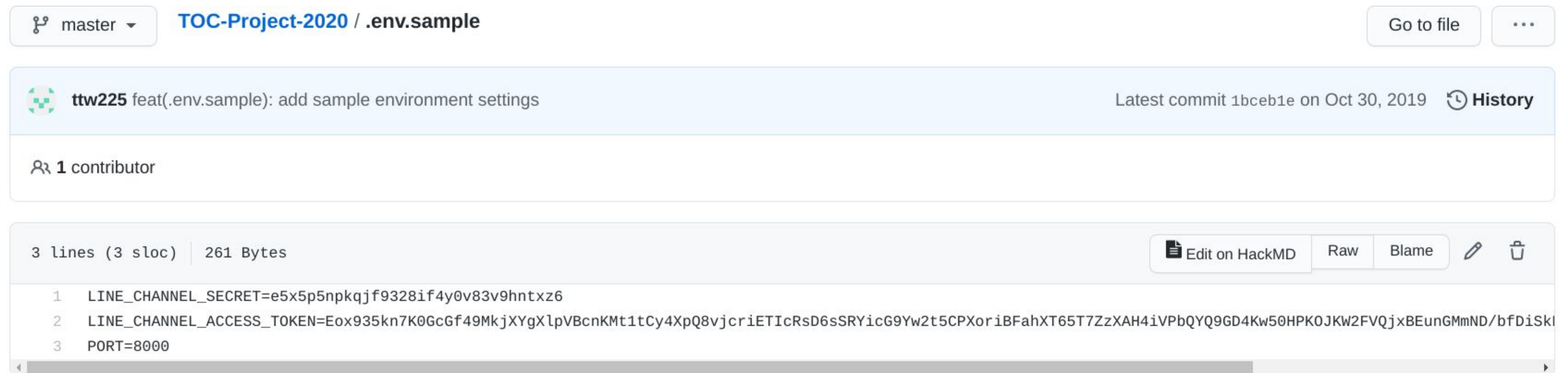
A - denoted as project-level python environment

Writing a HTTP server - LINE Token

24

To make Bot Server able to communicate with your channel, you need to set `LINE_CHANNEL_SECRET` and `LINE_CHANNEL_ACCESS_TOKEN` to your server environment, so the app have right permission to interact with your channel.

`LINE_CHANNEL_SECRET`, `LINE_CHANNEL_ACCESS_TOKEN` are obtained from previous steps (LINE channel creation)



The screenshot shows a GitHub repository interface for `TOC-Project-2020`. The file `.env.sample` is selected, showing its commit history and contributors. The file content is displayed in a code editor with line numbers 1 to 3. The content of the file is as follows:


```
1 LINE_CHANNEL_SECRET=e5x5p5npkqjf9328if4y0v83v9hntxz6
2 LINE_CHANNEL_ACCESS_TOKEN=Eox935kn7K0GcGf49MkjXYgXlpVBcnKmt1tCy4XpQ8vjcriETIcRsD6sSRYicG9Yw2t5CPXoriBFahXT65T7ZzXAH4iVPbQYQ9GD4Kw50HPK0JKW2FVQjxBEunGMmND/bfDiSkI
3 PORT=8000
```

Writing a HTTP server - Utils

25

utils.py - SDK for line message API

master [TOC-Project-2020 / utils.py](#) / <> Jump to

 **ttw225** feat(TOC-Demo): well done

1 contributor

23 lines (14 sloc) | 488 Bytes

```
1  import os
2
3  from linebot import LineBotApi, WebhookParser
4  from linebot.models import MessageEvent, TextMessage, TextSendMessage
5
6
7  channel_access_token = os.getenv("LINE_CHANNEL_ACCESS_TOKEN", None)
8
9
10 def send_text_message(reply_token, text):
11     line_bot_api = LineBotApi(channel_access_token)
12     line_bot_api.reply_message(reply_token, TextSendMessage(text=text))
13
14     return "OK"
15
16
17 """
18 def send_image_url(id, img_url):
19     pass
20
21 def send_button_message(id, text, buttons):
22     pass
23 """
```

Writing a HTTP server - FSM

26

fsm.py - building the fsm for LINE text message

master TOC-Project-2020 / fsm.py / <> Jump to

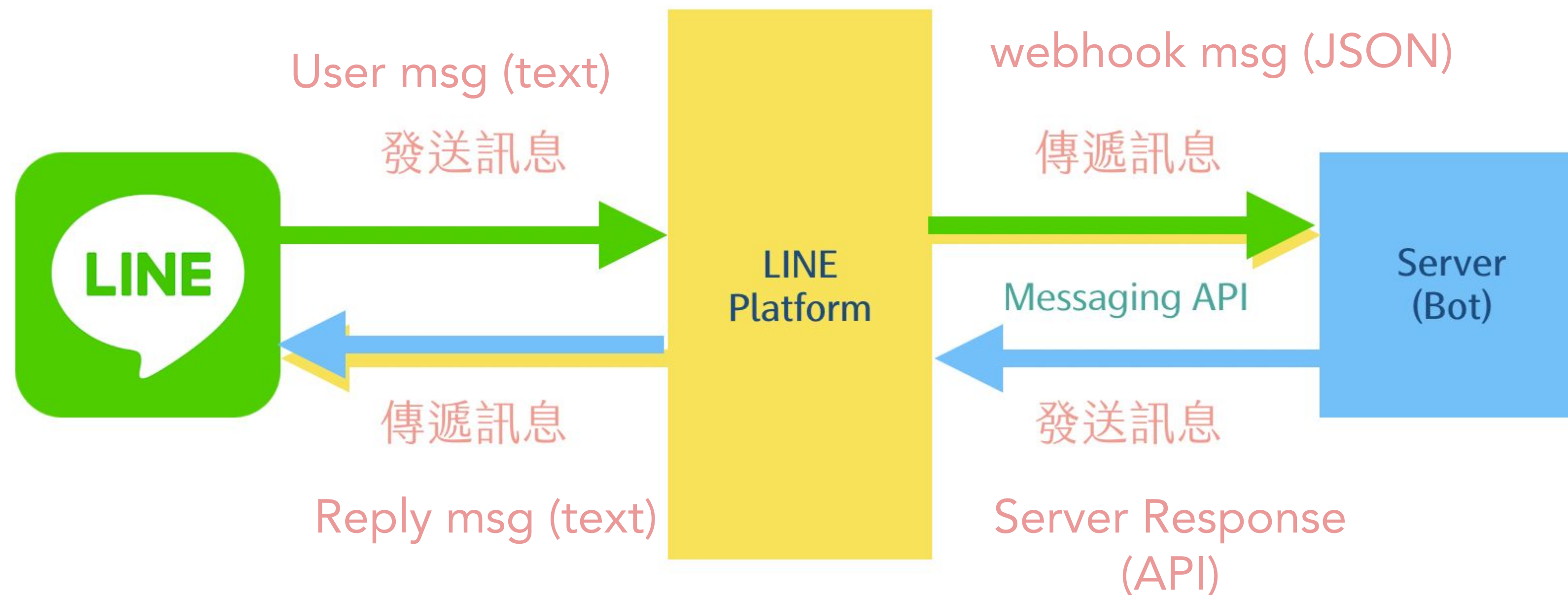
ttw225 feat(TOC-Demo): well done

1 contributor

36 lines (25 sloc) 993 Bytes

```
1 from transitions.extensions import GraphMachine
2
3 from utils import send_text_message
4
5
6 class TocMachine(GraphMachine):
7     def __init__(self, **machine_configs):
8         self.machine = GraphMachine(model=self, **machine_configs)
9
10    def is_going_to_state1(self, event):
11        text = event.message.text
12        return text.lower() == "go to state1"
13
14    def is_going_to_state2(self, event):
15        text = event.message.text
16        return text.lower() == "go to state2"
17
18    def on_enter_state1(self, event):
19        print("I'm entering state1")
20
21        reply_token = event.reply_token
22        send_text_message(reply_token, "Trigger state1")
23        self.go_back()
```


Below shows API call underneath the system. Main work of this project is to build the Server (bot).



▶ Writing a HTTP server - Clone & Trace

28

1. Clone the project
2. Build environment
3. Trace the code
4. [Refer the official documentation](#)
5. Extend your bot

▶ Test server locally

29

[ngrok](#) - secure introspectable tunnels to localhost

- Map your `localhost:8000` to `https domain`

Extra reference:

[ngrok reference](#)

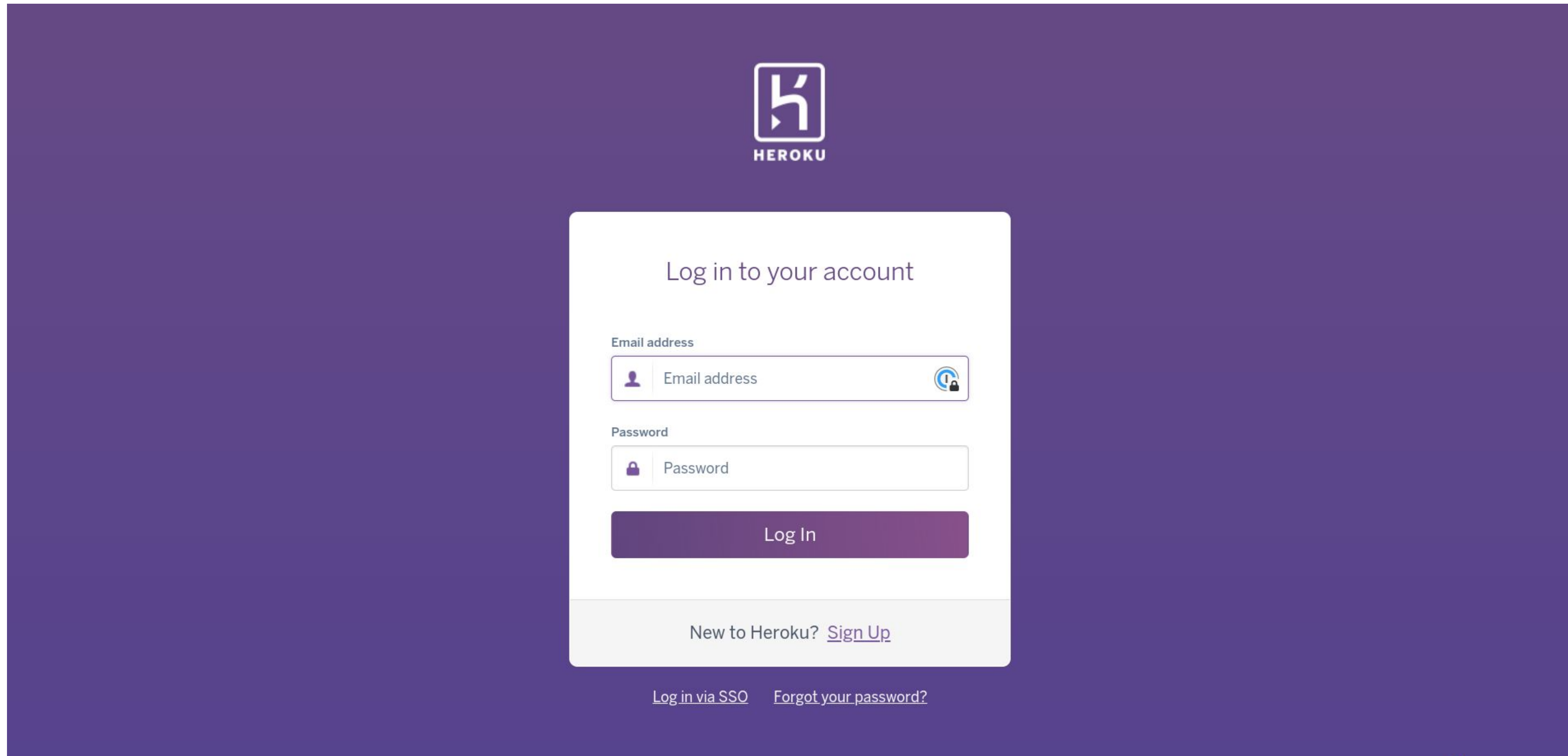
Web deployment is the process of deploying the code (html, css, javascript and server code) from source control or source artifacts to a hosting platform. This is usually in the cloud or on a local server.

Method and Choice

- A local machine with public static IP
- Cloud services
 - AWS - Amazon Web Service
 - Heroku - PaaS (Platform as a service)
 - Microsoft Azure
 - GCP - Google cloud platform

Deploy with Heroku

Register Heroku and create an app



The image shows the Heroku login page. At the top center is the Heroku logo, which consists of a stylized 'h' inside a square, with the word 'HEROKU' underneath. Below the logo is a white rectangular box with rounded corners. Inside this box, the text 'Log in to your account' is centered. Below this text are two input fields. The first field is labeled 'Email address' and contains a placeholder text 'Email address'. The second field is labeled 'Password' and contains a placeholder text 'Password'. Below these fields is a purple button with the text 'Log In'. At the bottom of the white box, there is a link that says 'New to Heroku? Sign Up'. Below the white box, there are two links: 'Log in via SSO' and 'Forgot your password?'.

HEROKU

Log in to your account

Email address

Email address

Password

Password

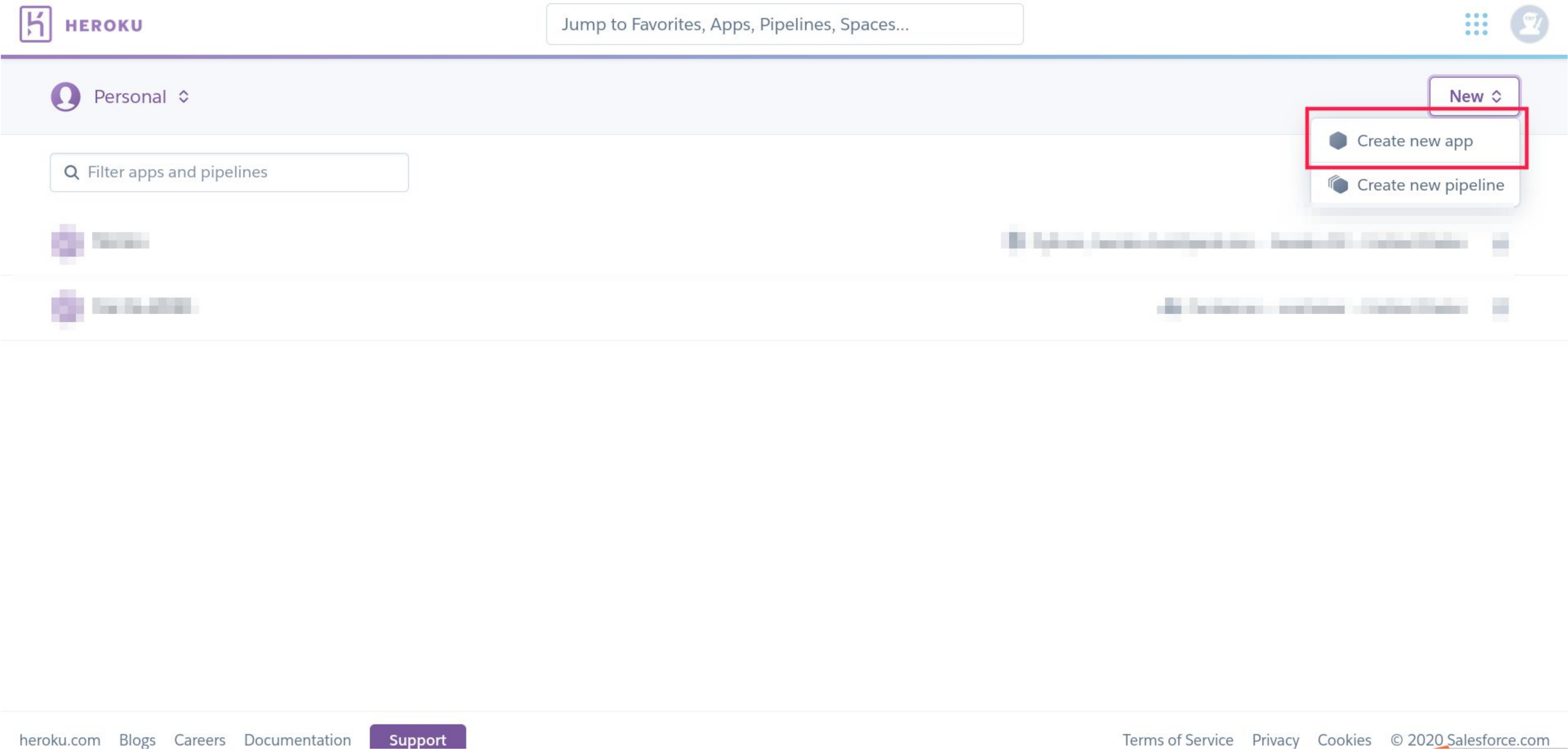
Log In

New to Heroku? [Sign Up](#)


[Log in via SSO](#) [Forgot your password?](#)

Deploy with Heroku



Register Heroku and create an app



Fill in app name



Jump to Favorites, Apps, Pipelines, Spaces...





Create New App

App name

app-name

Choose a region

 United States

 Add to pipeline...

Create app

heroku.com Blogs Careers Documentation **Support**

Terms of Service Privacy Cookies © 2020 Salesforce.com

[Install Heroku-Cli](#)

macOS

```
$ brew tap heroku/brew && brew install heroku
```

Windows

Download the appropriate installer for your Windows installation:

64-bit installer

32-bit installer

Ubuntu 16+

Run the following from your terminal:

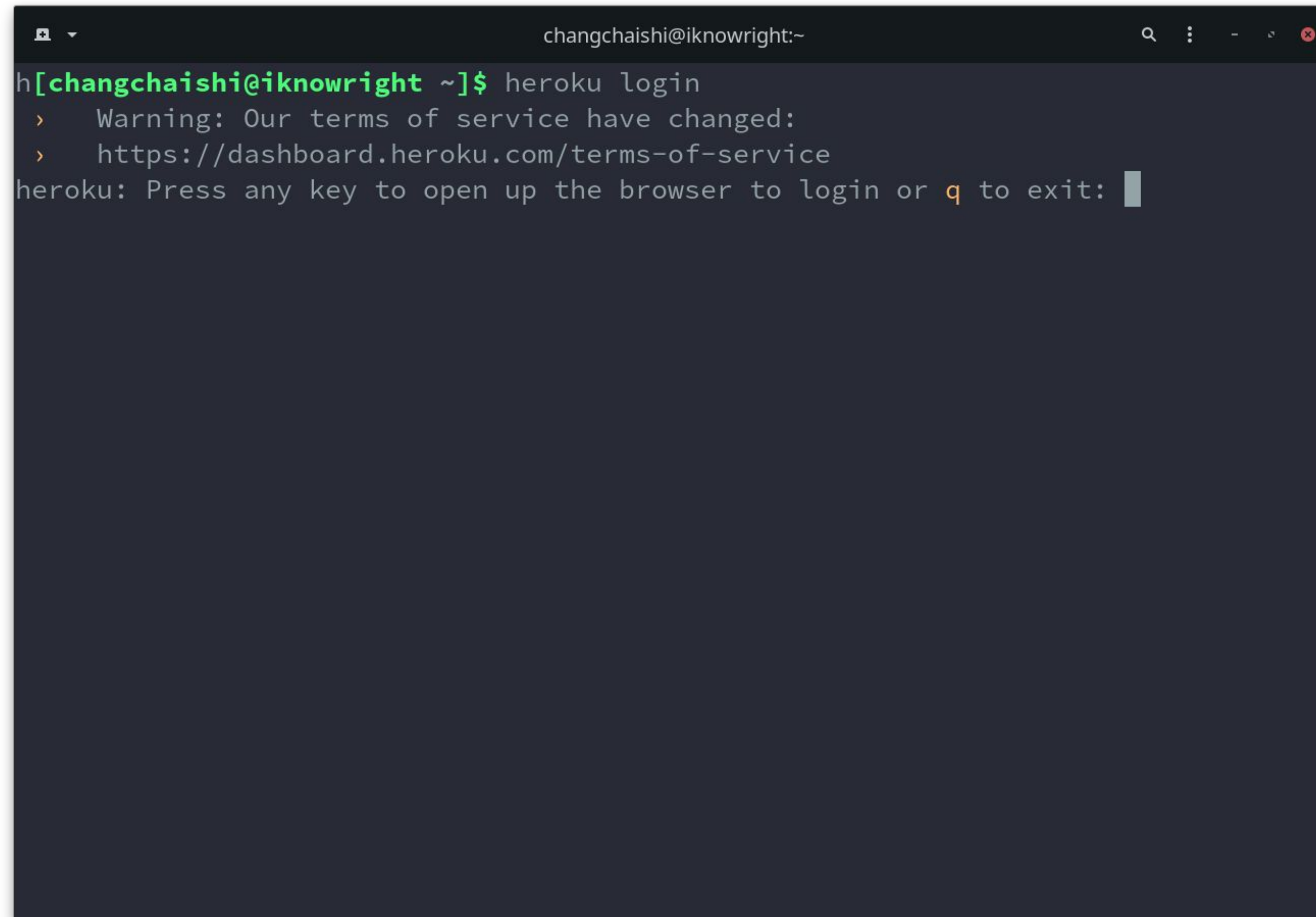
```
$ sudo snap install --classic heroku
```

[Snap is available on other Linux OS's as well.](#)

Deploy with Heroku

Login Heroku-Cli (Terminal depends on your OS)

- Terminal will redirect you to browser-login

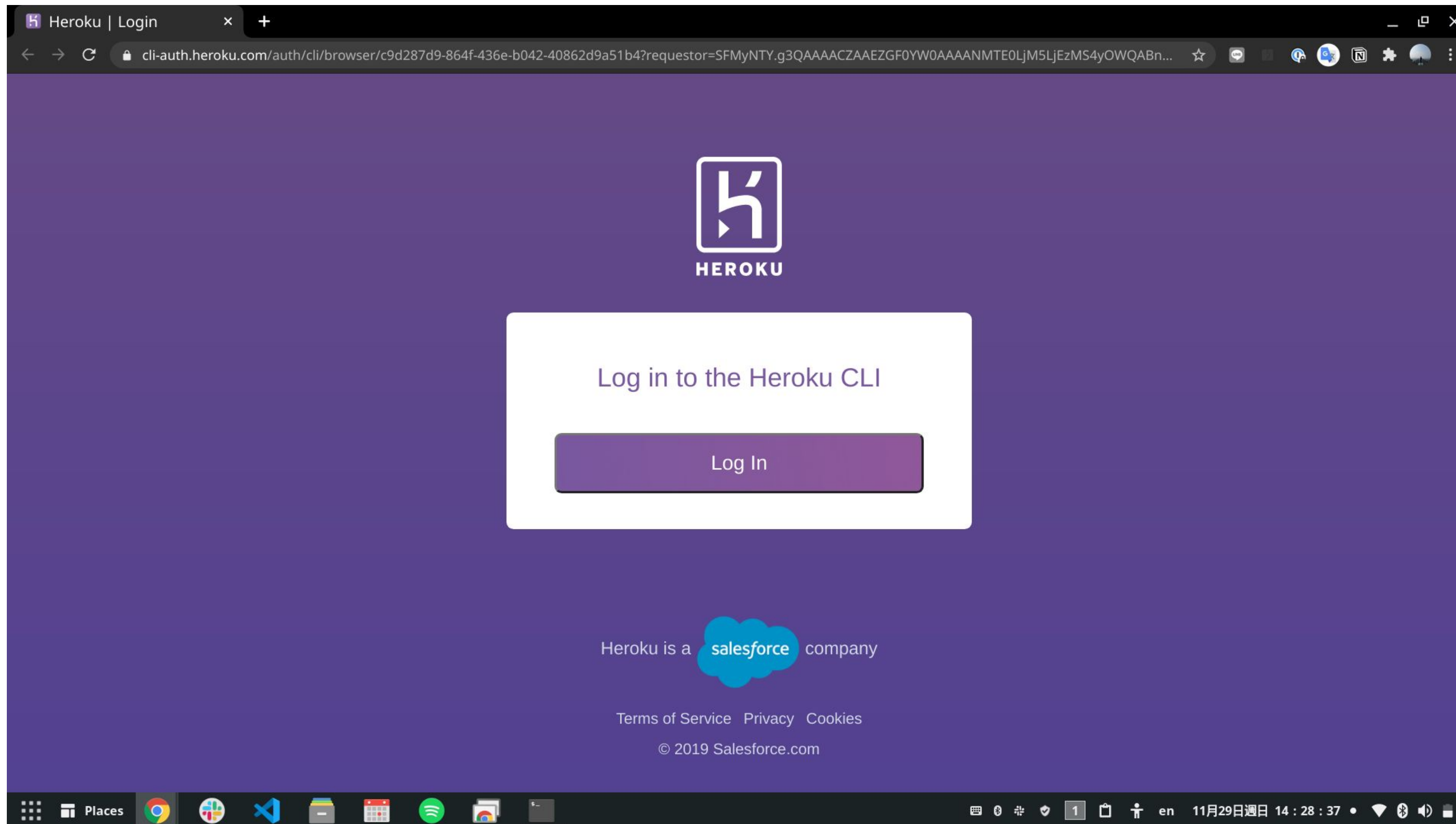
A terminal window with a dark background and light green text. The window title bar shows 'changchaishi@iknowright:~'. The prompt is 'h[changchaishi@iknowright ~]\$'. The user has entered 'heroku login'. The output shows a warning about updated terms of service with a link to 'https://dashboard.heroku.com/terms-of-service'. The prompt now says 'heroku: Press any key to open up the browser to login or q to exit:'.

```
changchaishi@iknowright:~  
h[changchaishi@iknowright ~]$ heroku login  
> Warning: Our terms of service have changed:  
> https://dashboard.heroku.com/terms-of-service  
heroku: Press any key to open up the browser to login or q to exit:
```

Deploy with Heroku

36

- Login Heroku-Cli (Terminal depends on your OS)
- Terminal will redirect you to browser-login



Deploy with Heroku

37

- Back to the CLI, you can see the logged in info.

```
changchaishi@iknowright:~  
h[changchaishi@iknowright ~]$ heroku login  
  > Warning: Our terms of service have changed:  
  > https://dashboard.heroku.com/terms-of-service  
heroku: Press any key to open up the browser to login or q to exit:  
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/c9d287d9-864f-43  
6e-b042-40862d9a51b4?requestor=SFMyNTY.g3QAAACZAAEZGF0YW0AAAANMTE0LjM5LjEzMS4yO  
WQABnNpZ25lZG4GAB4hsBJ2AQ.BV7qi4DAoiBUxYLY8sJXB1xn0sZko1cnHBTNfuDP70Y  
Logging in... done  
Logged in as changchaishi@gmail.com  
[changchaishi@iknowright ~]$
```

Using Git:

- What is git? Version control system
- Difference between git and github
- Local vs Remote
- Github vs Gitlab vs Heroku

Heroku: you can choose between

- Github
- Heroku
- Docker registry



Heroku Git
Use Heroku CLI



GitHub
Connected



Container Registry
Use Heroku CLI


```
[/your_project_dir] $ heroku git:remote -a {HEROKU_APP_NAME}
```

```
[/your_project_dir] $ git add .
```

```
[/your_project_dir] $ git commit -m "commit msg"
```

```
[/your_project_dir] $ git push -f heroku master
```

```
[/your_project_dir] $ Heroku logs --tail -a {HEROKU_APP_NAME}
```

After pushing your code to heroku, heroku will automatically build your app, use heroku cli to check build logs

Binding Deployed APP to Line Console

40

Tell Line which webhook url to POST. (Your server url).

LINE Developers

ProductsDocumentsAPI referenceNewsFAQCommunityBlog

Console home

Providers

Search...

Admin

Notifier

TOC-TA

計理助教

Tools

Support

TOP > TOC-TA > NCKU-TOC-TA > Messaging API

Webhook settings

Webhook URL ?https://toc-ta-2021.herokuapp.com/

VerifyEdit

Use webhook ?☒

LINE Official Account features

Edit the message text and other settings for these features in the LINE Official Account Manager

- Basic (60%)
- Present (10%)
- Functionality (10%)
- Creativity (10%)
- Bonus (5~10%)

- Able to show or generate FSM Diagram
- Bot running properly
- At least 3 states for FSM

- Demo Smoothness
- Documentation (README and etc.)
- Illustration

Functionality (10%)

- State complexity (Not amount of states)
- CRUD (create read update delete)
- Parsing

Creativity (10%)

Line Bot genre, theme

- Sports
- Game
- Service
- News
- Techs
- Tools

- Deploy
- Extra functionality or technics
 - image/sound/video
 - web crawling
 - binding database
 - machine learning
 - blockchain
 - others

[台灣事實查核中心](#)

[疾管家](#)

[Cofacts 真的假的](#)

[Line Developer 註冊](#)

[LINE 開發環境](#)

[Heroku](#)

[LineBot + python + Heroku](#)

[ngrok](#)

Bot Platform: Line (exclusively)

Programming Language / Framework: Unrestricted

- node.js
- golang
- anything else that supports FSM graph

LINE

- Line Reply Token - can only reply once
- Line Bot (Free ver.) - friend limit: 50 people

Heroku

- Heroku - dyno will be in sleep mode when no activities found in the server

主旨:[Final Project] 簡述遇到的問題

內文:

- 姓名學號:
- 系統環境:
- 系統版本:
- 套件版本:
- 在哪一個步驟遇到的問題:
- 詳述問題:
- 完整的錯誤訊息:
- 已經試過的解決方法:
- 在這個問題上已經花費的時間:

[2018 FAQ](#)

Thanks