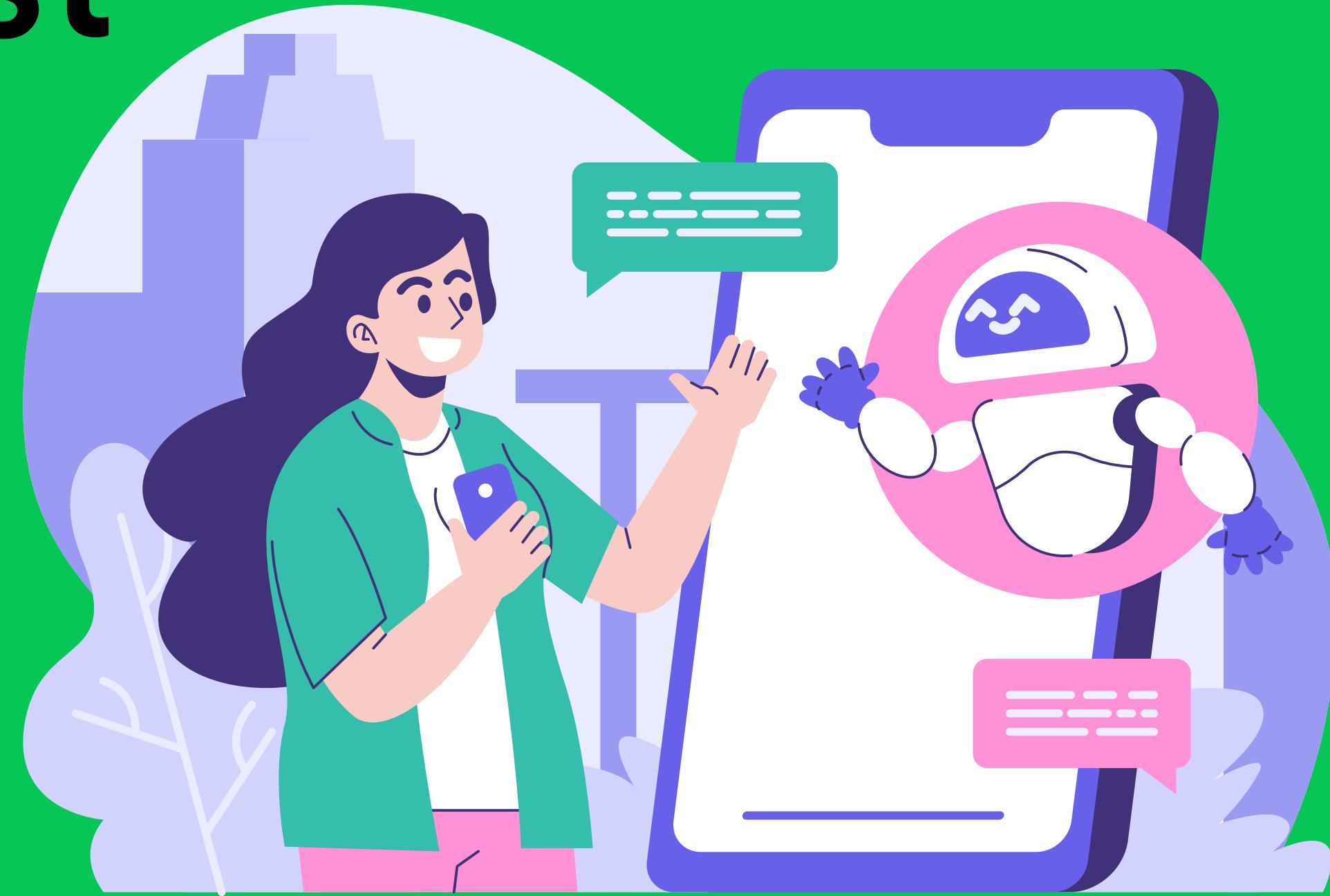


Build your first LINE chatbot



le wagon

What you will make



Read 10:24 PM What shall I eat?

spaghetti 10:24 PM

Read 10:24 PM How's the weather in London?

The weather is mostly CLOUDS in London for the next 4 days.
Except on tomorrow(Rain), Saturday(Rain).

The temperature will be:
9°C for today
5°C for tomorrow
7°C for Saturday
9°C for Sunday

10:24 PM

> W | >

W | What | We

1 2 3 4 5 6 7 8 9 0

q w e r t y j o p

a s d f g h k l

z x c v b n m

?123 , . English ←

A screenshot of a mobile messaging app showing a conversation with a bot. The user asks "What shall I eat?", and the bot replies with "spaghetti". The user then asks "How's the weather in London?", and the bot provides a detailed weather forecast for the next four days, including temperatures and precipitation. Below the messages is a virtual keyboard.

Answer simple
questions

Tells you the
weather

Detect what's in the
picture

and more....

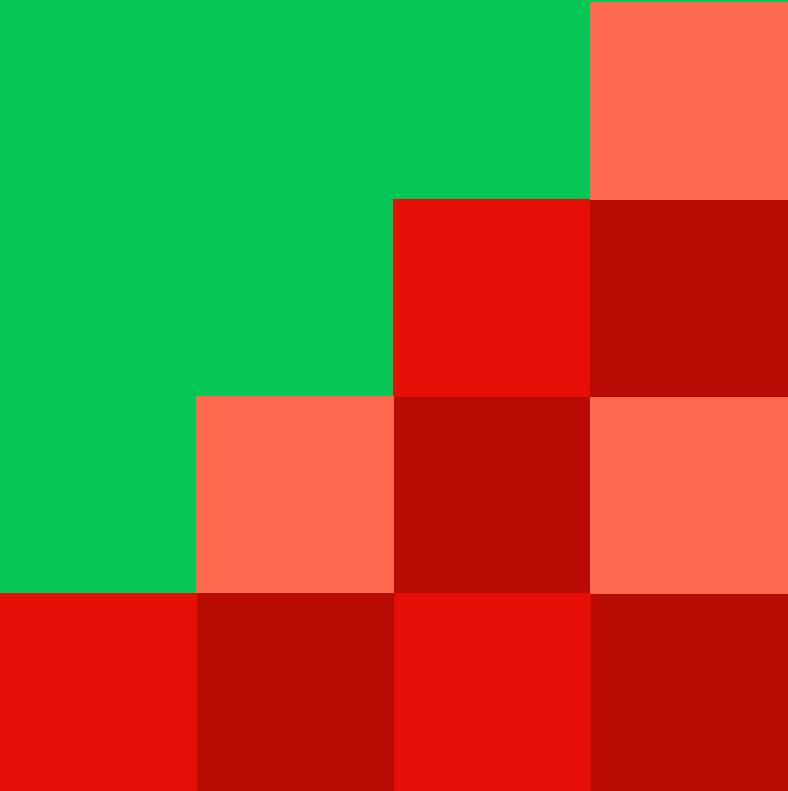
What you will learn



1. How to code logic using if-else statements to handle responses and decision-making.
2. How to leverage the powers of APIs to make your chatbot do more than saying things back
3. Deploying to the cloud, understanding how webhooks and services communicate in a live app.

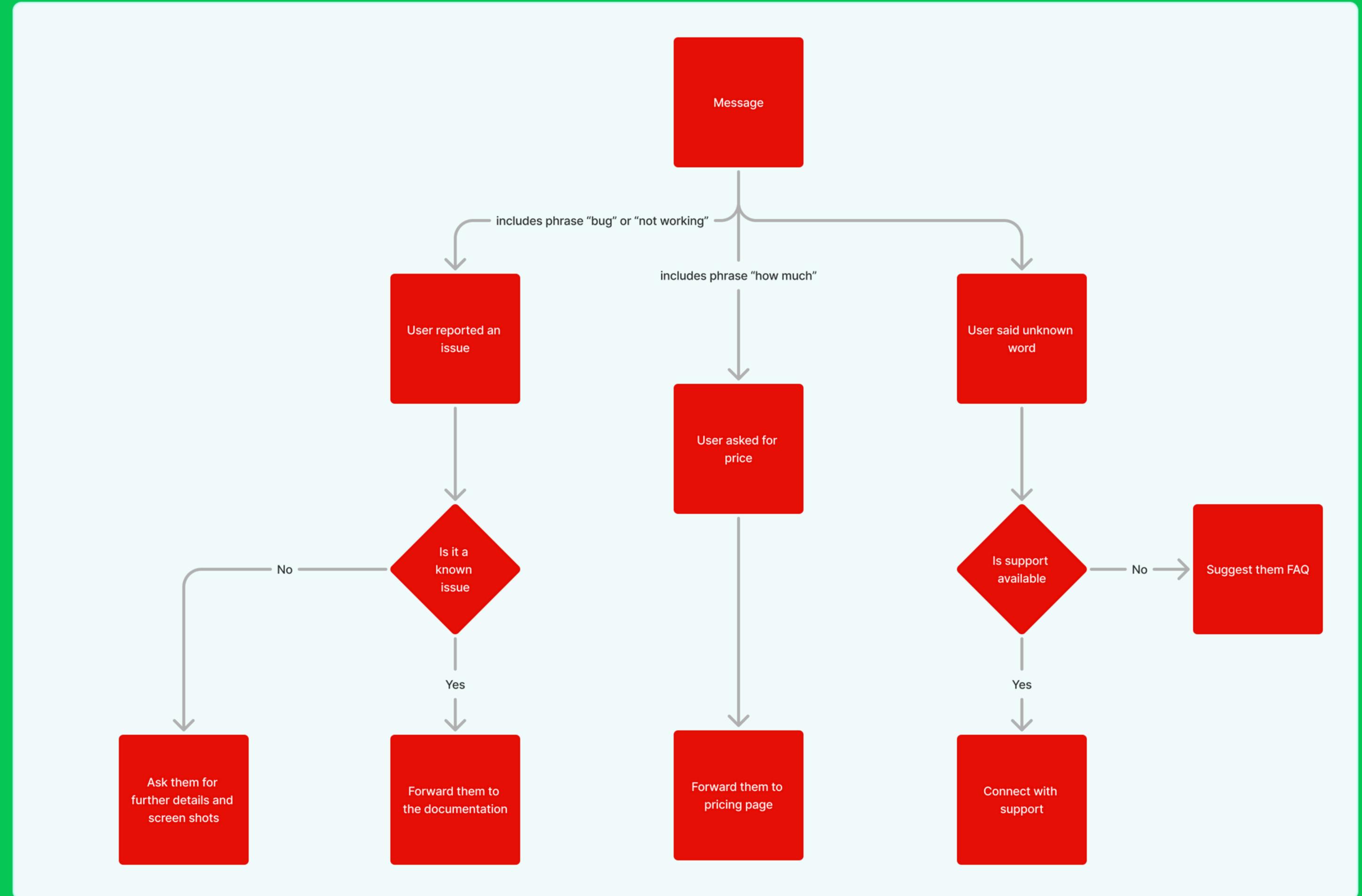
Workshop steps

1. Discuss how the chatbot makes **decisions** 
2. Code together the **brain** of the bot 
3. Set things up on LINE and **deploy** 



How does chat bot make decisions?

Decision tree



Conditional statement

if the user says hello

the bot responds with hello

elseif the user asks for the weather

the bot responds with weather forecast

else

the bot says it doesn't know that

end



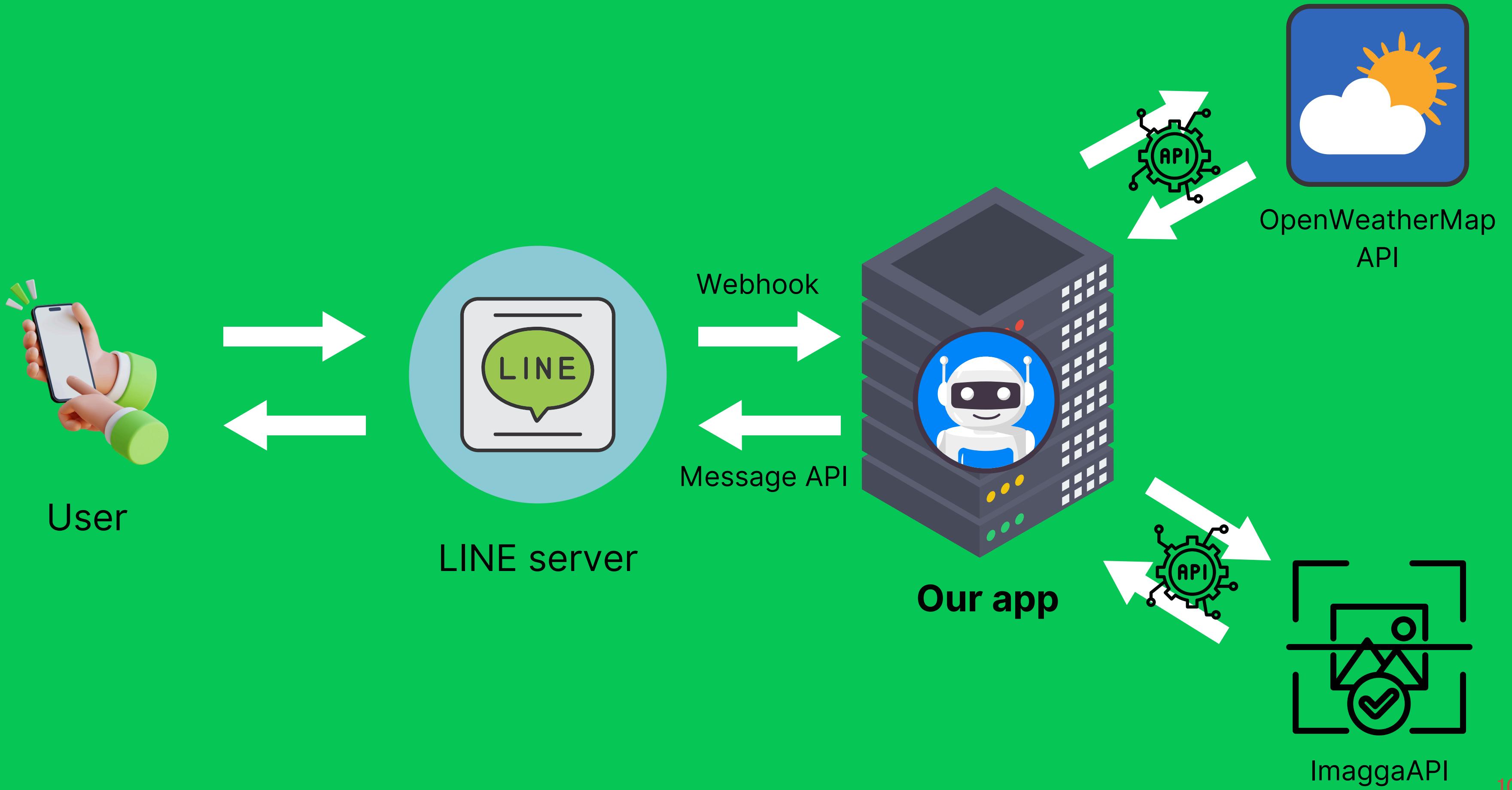
Let's build the brain with ruby!

Using repl.it/languages/Ruby



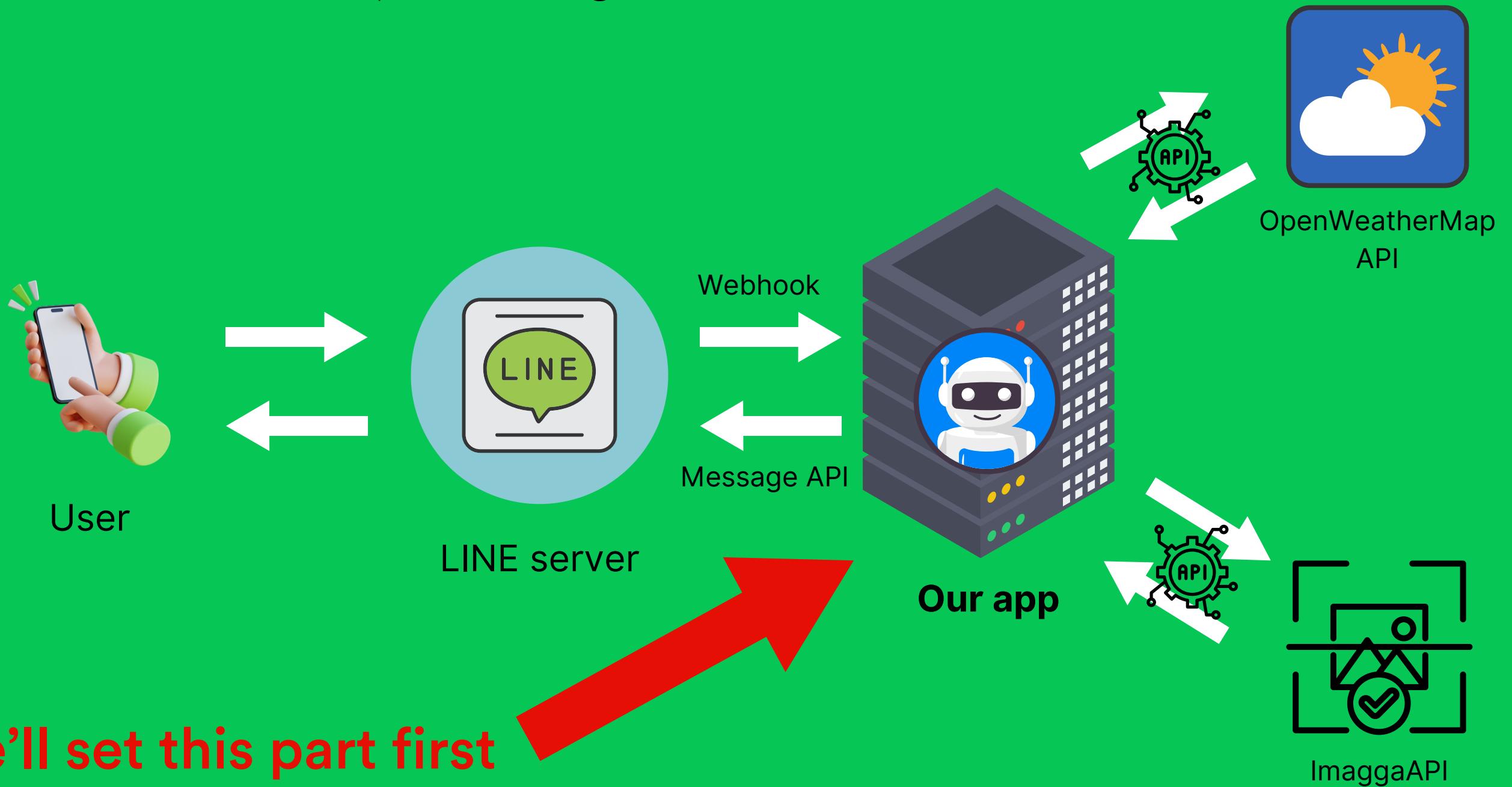
Now the logic is there, but how
do we make that talk on **LINE??**





Let's deploy it to the cloud ☁

So we don't have to have our computer running 24/7



Now, turn the logic to the **bot**



The logic we just coded



Sinatra
<https://sinatrarb.com>

**Sinatra is an open-source Ruby framework for building web applications.
It helps us handle web requests.**

Check out the Repo

<https://github.com/SuzukiRyuichiro/bob-the-bot-v4>

SuzukiRyuichiro/**bob-the-bot-v4**

A Line chatbot for Le Wagon workshop
(Sinatra/Line API/Googel Gemini API)

0 Contributors 0 Issues 0 Stars 0 Forks

SuzukiRyuichiro/bob-the-bot-v4: A Line chatbot for Le Wagon workshop (Sinatra/Line API/Googel Gemini API)

A Line chatbot for Le Wagon workshop (Sinatra/Line API/Googel Gemini API) - SuzukiRyuichiro/bob-the-bot-v4

 GitHub

Copy the project and deploy



1. Clone the app from Github

```
git clone https://github.com/SuzukiRyuichiro/bob-the-bot-v4.git [CHANGE_THIS_TO_YOUR_BOT_NAME]
```

2. Move into the folder

```
cd [CHANGE_THIS_TO_YOUR_BOT_NAME]
```

3. Login to Heroku

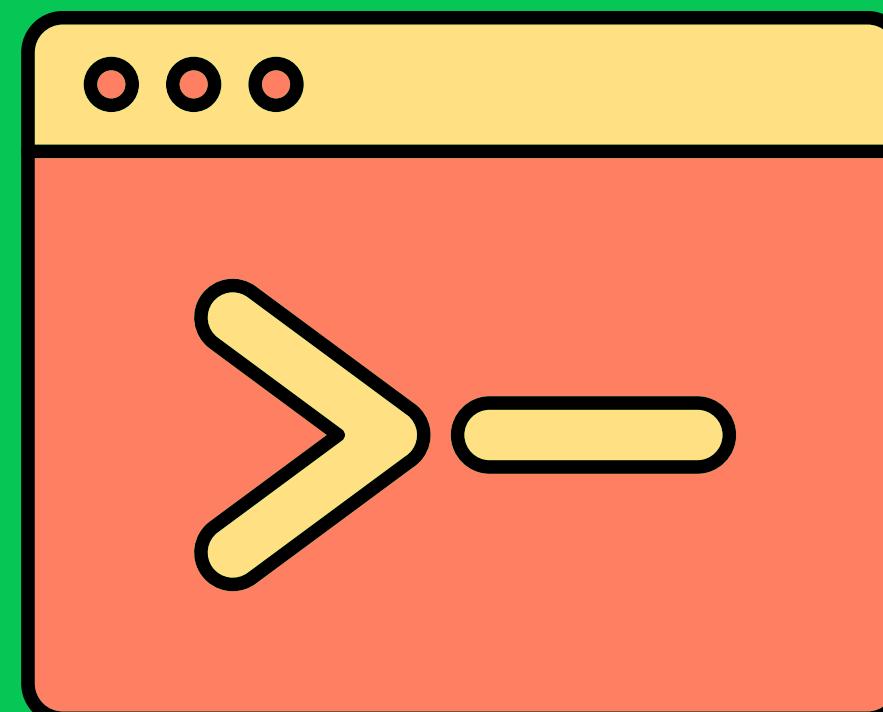
```
heroku login
```

4. Create your Heroku app

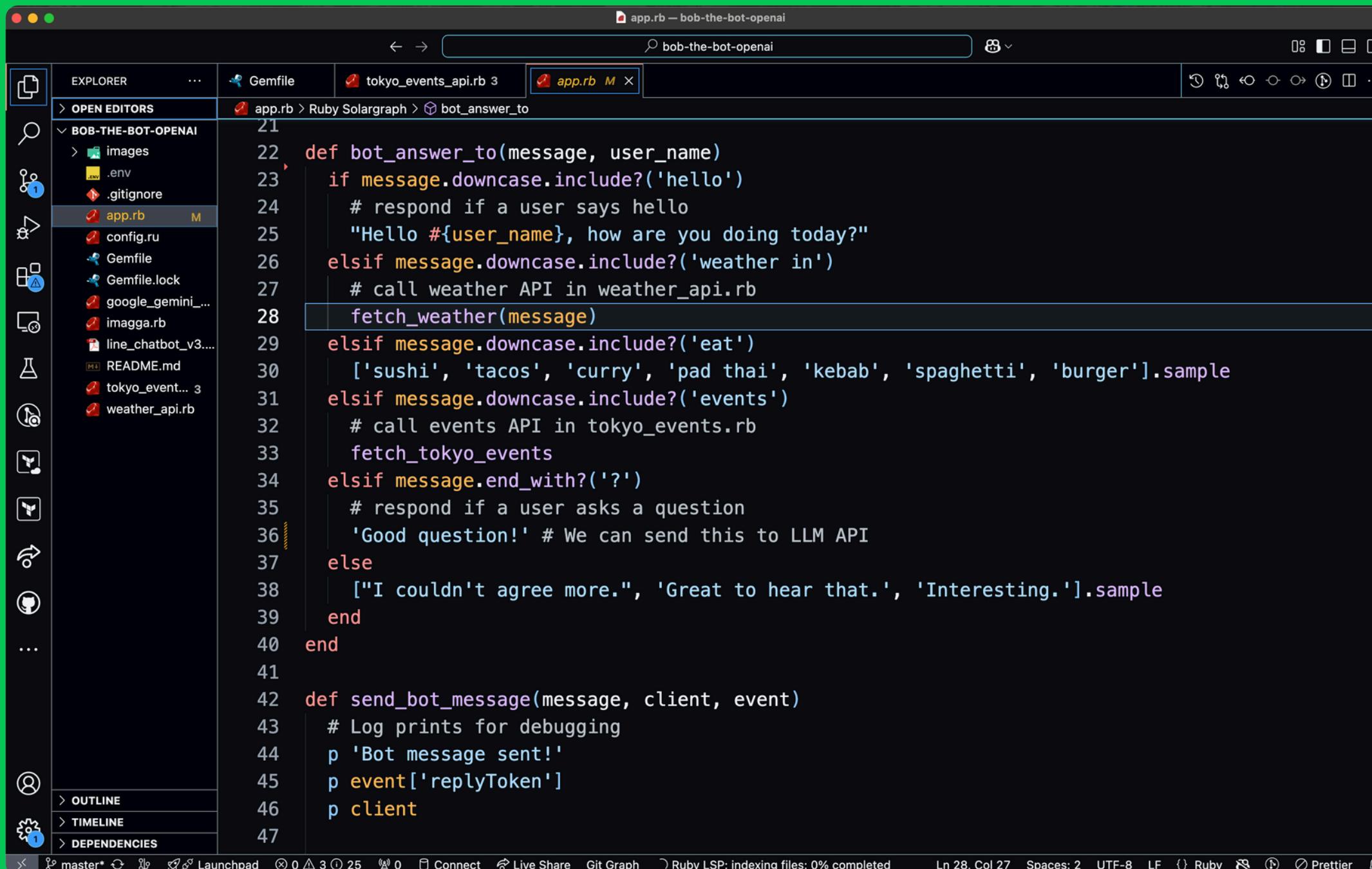
```
heroku create [CHANGE_THIS_TO_YOUR_BOT_NAME]
```

5. Push the code to Heroku

```
git push heroku master
```



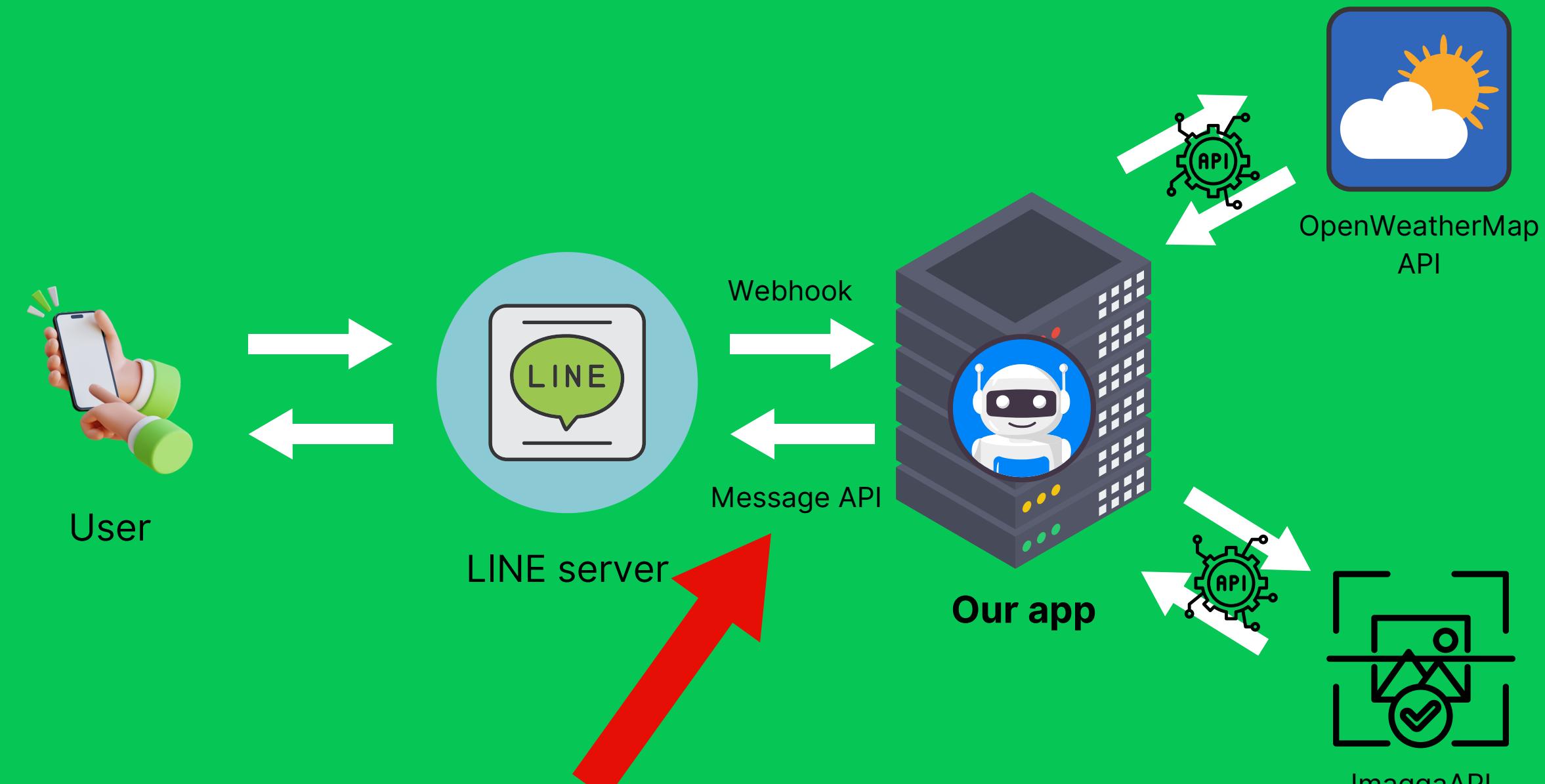
Find the brain of the bot



The screenshot shows a code editor window titled "app.rb — bob-the-bot-openai". The editor displays the contents of the "app.rb" file within the "BOB-THE-BOT-OPENAI" project. The code is written in Ruby and defines two methods: "bot_answer_to" and "send_bot_message". The "bot_answer_to" method handles various user messages by checking if they include specific keywords like "hello", "weather", "eat", or "events". It also checks if a message ends with a question mark. The "send_bot_message" method logs the message sent to the client. The code editor interface includes a sidebar for navigation and a bottom status bar showing file paths and system information.

```
21
22 def bot_answer_to(message, user_name)
23   if message.downcase.include?('hello')
24     # respond if a user says hello
25     "Hello #{user_name}, how are you doing today?"
26   elsif message.downcase.include?('weather in')
27     # call weather API in weather_api.rb
28     fetch_weather(message)
29   elsif message.downcase.include?('eat')
30     ['sushi', 'tacos', 'curry', 'pad thai', 'kebab', 'spaghetti', 'burger'].sample
31   elsif message.downcase.include?('events')
32     # call events API in tokyo_events.rb
33     fetch_tokyo_events
34   elsif message.end_with?('?')
35     # respond if a user asks a question
36     'Good question!' # We can send this to LLM API
37   else
38     ["I couldn't agree more.", 'Great to hear that.', 'Interesting.'].sample
39   end
40 end
41
42 def send_bot_message(message, client, event)
43   # Log prints for debugging
44   p 'Bot message sent!'
45   p event['replyToken']
46   p client
47
```

Set things up in LINE



We have to tell LINE about our app



1. Get the app and sign up fro the app
2. Go to <https://account.line.biz/signup> and signup for biz account
3. Login to biz account

LINE Business ID

[Sign up with LINE account](#)

[Sign up with email](#)

[Already have an account?](#)

By logging in to LINE Business ID, you agree to the [Terms of Use](#).

[>About LINE Business ID](#)

[English ▾](#)

[Help](#) [Terms of Use](#) © LY Corporation



TOP

Recently visited channels



Le wagon bot

Messaging API



鈴木龍一郎

Messaging API

Providers (3)

Create



Provider	Role
lewagon	Admin
Scooter	Admin
scooter_the_bot	Admin



TOP > lewagon2

lewagon2

Channels Roles Settings

This provider doesn't have any channels yet

To create one, choose a channel type below

A red arrow points from the text "To create one, choose a channel type below" to the central "Messaging API" icon.

Create a LINE Login channel Create a Messaging API channel Create a Blockchain Service channel



TOP > **Create a new channel**

Create a Messaging API channel

It's no longer possible to create Messaging API channels directly from the LINE Developers Console.

To create a Messaging API channel, create a LINE Official Account using the Create a LINE Official Account button below, and then enable the use of the Messaging API on the LINE Official Account Manager.

For more information, see [Get started with the Messaging API](#) in the Messaging API documentation.

Create a LINE Official Account
(Go to the external site)





1 会社・店舗情報を登録 2 入力内容の確認 3 申し込み完了

LINE公式アカウントの作成

● 必須

ログイン情報

ユーザー名 [ログアウト](#)

サービス対象国・地域 日本 [日本の料金プランが適用されます。](#)

アカウント情報

アカウント名 • 例: Brownカフェ 0/20
LINEの友だちリストやトーク画面に表示される名称です。

メールアドレス • 0/240

会社・事業者の所在国・地域 • ② 日本
ここで設定した国・地域は、ユーザーが閲覧できるページ（アカウントのプロフィールなど）に表示されます。

会社・事業者名 0/100

業種 • 大業種を選択 小業種を選択 0/100

運用目的（複数選択可） •
 お店を知ってもらいたい、興味を持ってもらいたい
 来店してもらいたい、リピーターになってもらいたい

1 会社・店舗情報を登録 2 入力内容の確認 3 申し込み完了

入力内容の確認

アカウント情報

アカウント名 Le wagon the bot 2
メールアドレス [確認用アドレス](#)

会社・事業者の所在国・地域 日本
会社・事業者名

業種 ファッション・着物・呉服
運用目的（複数選択可） その他
主な使い方 未定

公式アカウントの運用に役立つ情報をお届けします

→ LINEヤフー for Businessを友だち追加
法人向けに、LINE公式アカウント、LINE広告、LINEミニアプリなどに役立つ情報を配信。

キャンセル 完了

*作成したアカウントは自動的にLINEの友だちに追加されます



The official account will be added to your friend automatically



A blue circular profile icon with a white person icon and a small edit pencil icon.

Le wagon the bot 2

1 · @221rxlar

[Edit profile](#)

Broadcasts

Broadcast messages to friends.

A paper airplane icon.

[Send a broadcast](#) [Use a template](#)

Content

Create rich messages, coupons, and more for your broadcasts.

A speech bubble icon with the word "LINE" inside.

[Create content](#)



LINE Official Account Manager Le wagon the bot @221rxlar (コミュニケーション) 1 Chat : Off Scooter_Ryuic... Help

Home Insights Chats Profile LINE VOOM Extensions Monetization Settings

Settings

- Account settings
- Manage permissions
- Response settings
- Messaging API** ←
- Registered info

Activity and billing

- Dashboard
- Monthly plan
- Premium ID
- Billing history
- Payment method
- Campaign codes
- Invoice info

Earnings

- Dashboard

Messaging API

Messaging API is an advanced feature for developers. It allows accounts to promote more interactive communication by sending and receiving messages and actions via the API.

[What is Messaging API?](#)
[LINE Developers API documentation](#)

Status: **Disabled**

Enable Messaging API ←

Hide menu labels © LY Corporation Terms and policies

Select provider

Please select the company or person who operates this account.

A provider is an individual developer, company, or organization that manages users' personal information to offer various services.
More details at [LINE Developers](#).

New provider
 0/100

scooter_the_bot

Scooter

lewagon

lewagon2 ←

By tapping Agree below, you agree to the [Messaging API Terms and Conditions of Service](#).



Privacy Policy and Terms of Use

Please enter the Privacy Policy and Terms of Use for the provider. You can edit this info later.

Privacy Policy 0/500
 This link is optional.

Terms of Use 0/500
 This link is optional.

Enabling Messaging API

You won't be able to change or unlink this provider once linked.

Enable Messaging API with the following info?

Account name Le wagon the bot

Provider name lewagon2



Messaging API

Messaging API is an advanced feature for developers. It allows accounts to promote more interactive communication by sending and receiving messages and actions via the API.

[What is Messaging API?](#)

[LINE Developers API documentation](#)

Change this to yours

Status Enabled

Channel info Channel ID

[Copy](#)

Channel secret

[Copy](#)

Webhook URL

[Your App URL]/callback

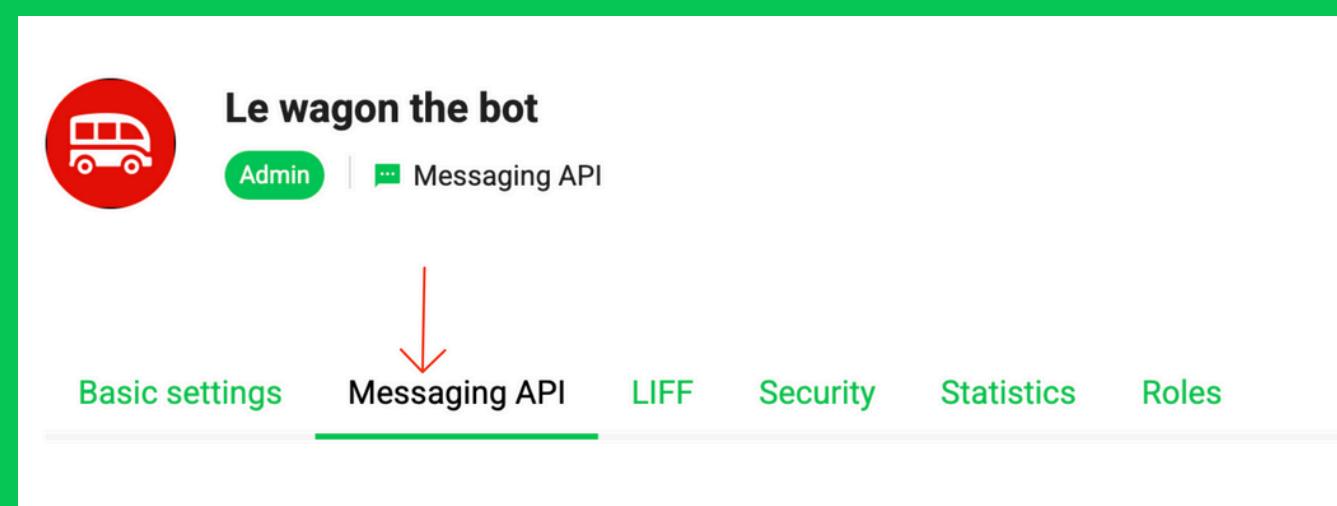
[Save](#)

Please enter a valid URL starting with "https://".



Now back at the developer console

The screenshot shows the LINE Developers console interface. On the left, there's a sidebar with sections for 'Console home', 'Providers' (with a search bar), 'Admin' (selected), and lists for 'lewagon', 'lewagon2' (highlighted with a red arrow), 'Scooter', and 'scooter_the_bot'. The main area shows a navigation bar with 'TOP > lewagon2'. Below it, the title 'lewagon2' is displayed with tabs for 'Channels' (selected), 'Roles', and 'Settings'. A large callout box with bold text 'You should have the official account you just created' has arrows pointing from its bottom right to the 'lewagon2' entry in the sidebar and to the 'Channels' tab. The 'Channels' section shows a card for 'Le wagon the bot' with a bus icon, labeled 'New Admin'. At the bottom, there are buttons for 'Create a new channel' and 'Messaging API'.



Le wagon the bot

Admin | Messaging API

Basic settings **Messaging API** LIFF Security Statistics Roles



Webhook settings

Webhook URL <https://bot-the-bot-4ce56bfc2763.herokuapp.com/callback>

Verify Edit This should be the URL you set up earlier

Use webhook ←

Channel access token

Channel access token (long-lived) [?](#)

Issue ←

Scroll down



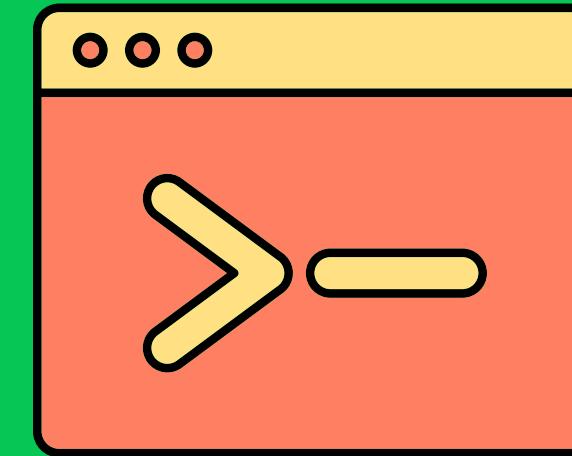
Copy the necessary credentials

1. Channel secret from the
basic settings

The screenshot shows the 'Basic Settings' page for a bot channel. It includes sections for App types (Bot), Permissions (PROFILE), Channel secret (redacted), Assertion Signing Key (Register a public key), and Your user ID (redacted). A red arrow points to the 'Channel secret' field.

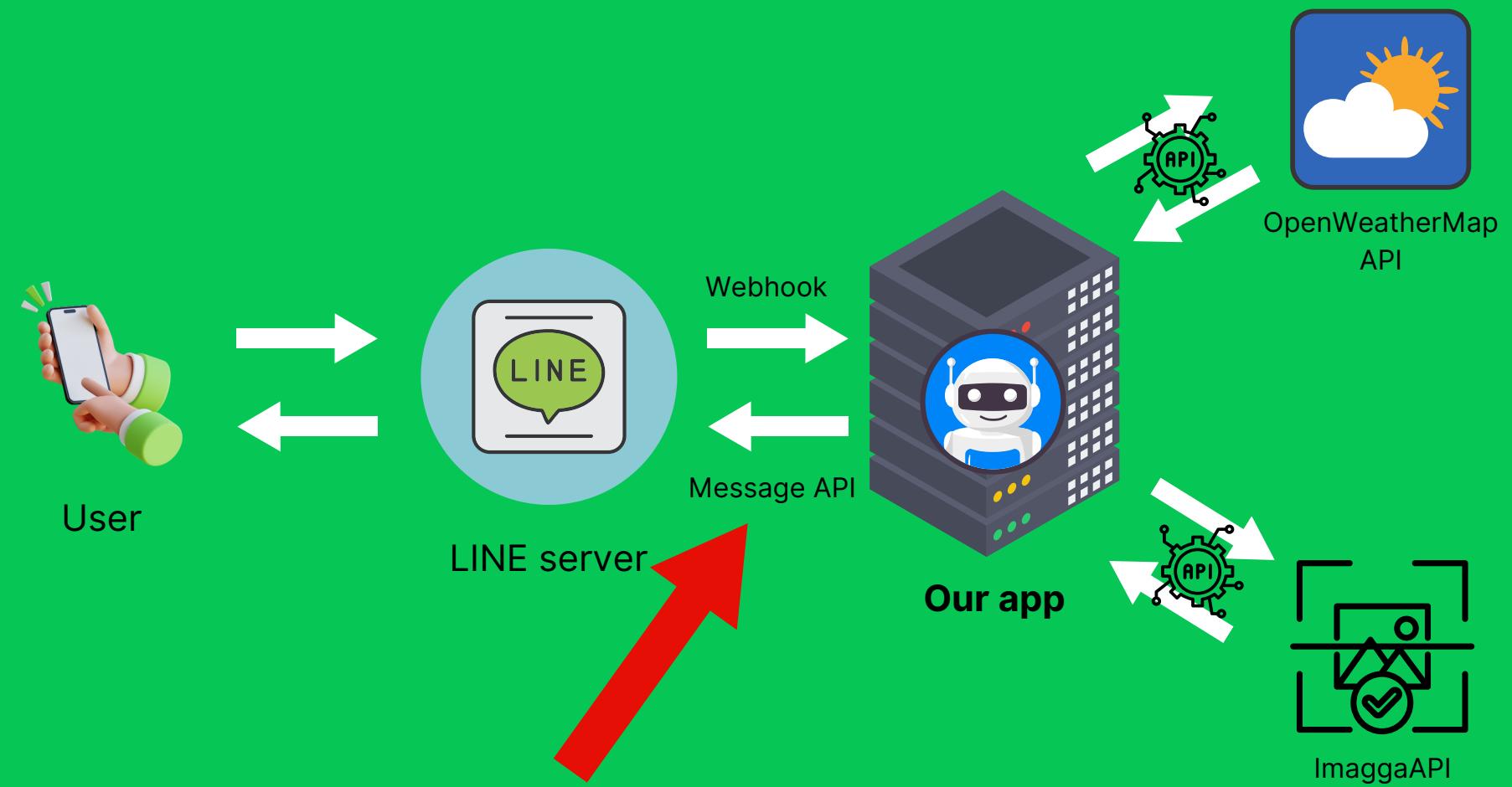
2. Channel access token
from the **Message API settings**

The screenshot shows the 'LINE Official Account features' page. It includes sections for Allow bot to join group chats (Disabled), Auto-reply messages (Enabled), Greeting messages (Enabled), and Channel access token (redacted). A red arrow points to the 'Channel access token' field.

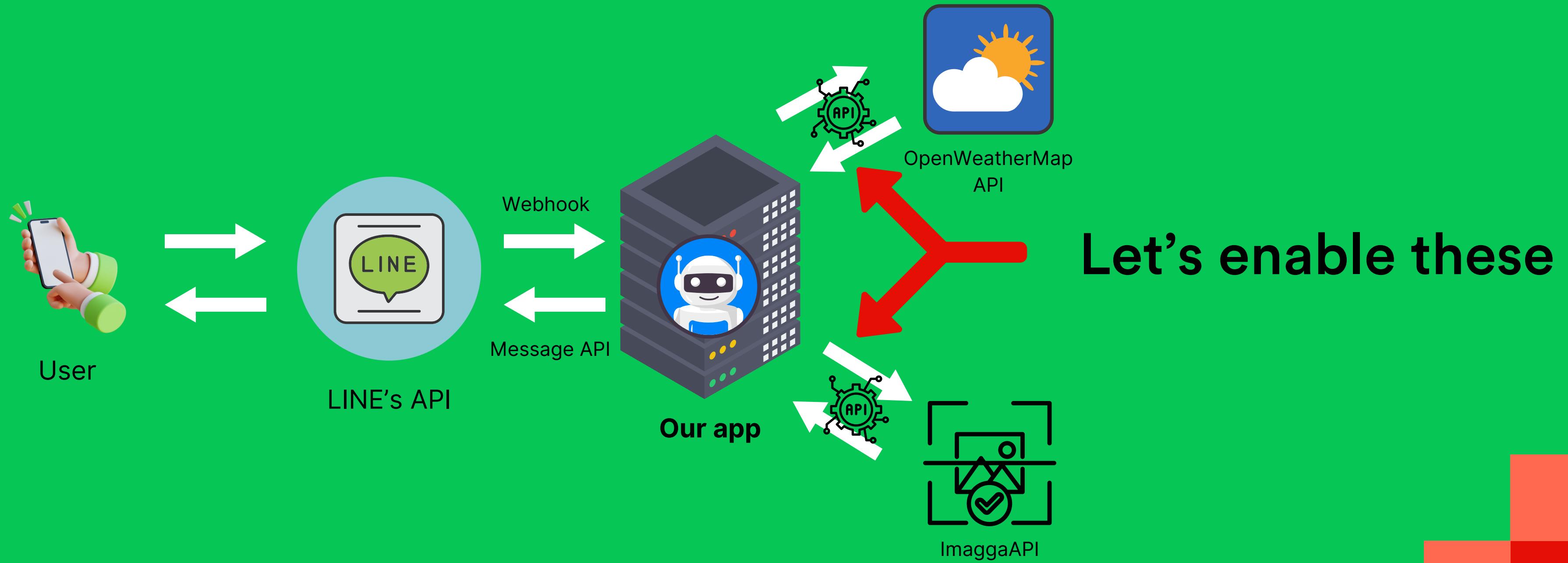


Tell Heroku about those credentials

```
heroku config:set LINE_CHANNEL_SECRET=[SECRET_YOU_COPIED]  
heroku config:set LINE_ACCESS_TOKEN=[TOKEN_YOU_COPIED]
```



Setting APIs up



OpenWeatherMap

https://home.openweathermap.org/users/sign_up

The screenshot shows the 'API keys' section of the OpenWeatherMap dashboard. At the top, there is a navigation bar with links: New Products, Services, API keys (which is the active tab), Billing plans, Payments, Block logs, My orders, My profile, and Ask a question. Below the navigation bar, a message states: 'You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.' A table lists existing API keys:

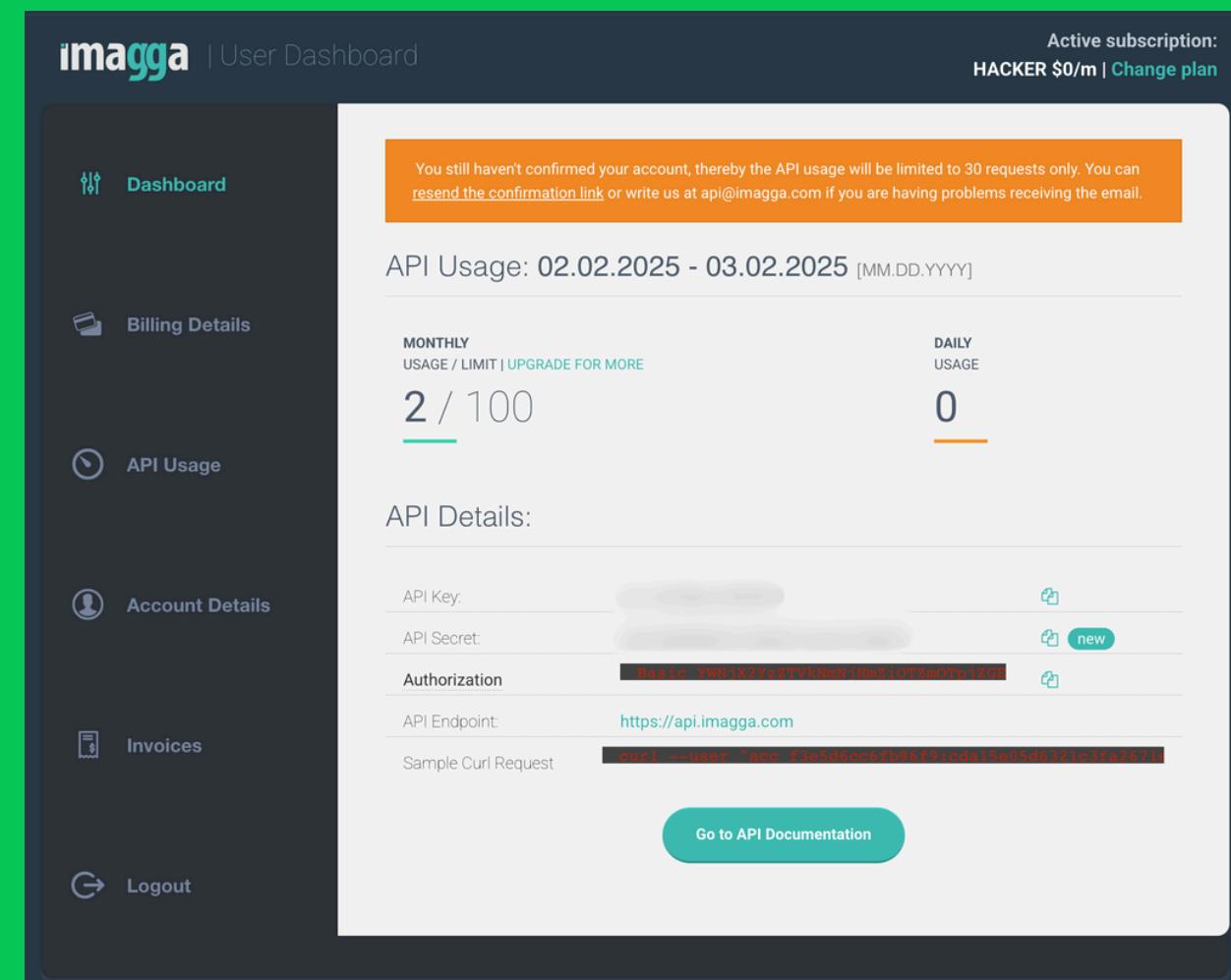
Key	Name	Status	Actions
[REDACTED]	Alfred	Active	[Switch] [Edit] [Delete]
[REDACTED]	line bot	Active	[Switch] [Edit] [Delete]

To the right of the table is a 'Create key' form with a text input field labeled 'API key name' and a 'Generate' button.

Copy the generated key and in terminal
heroku config:set WEATHER_API=[KEY_YOU_COPIED]

Imagga

<https://imagga.com/auth/signup>



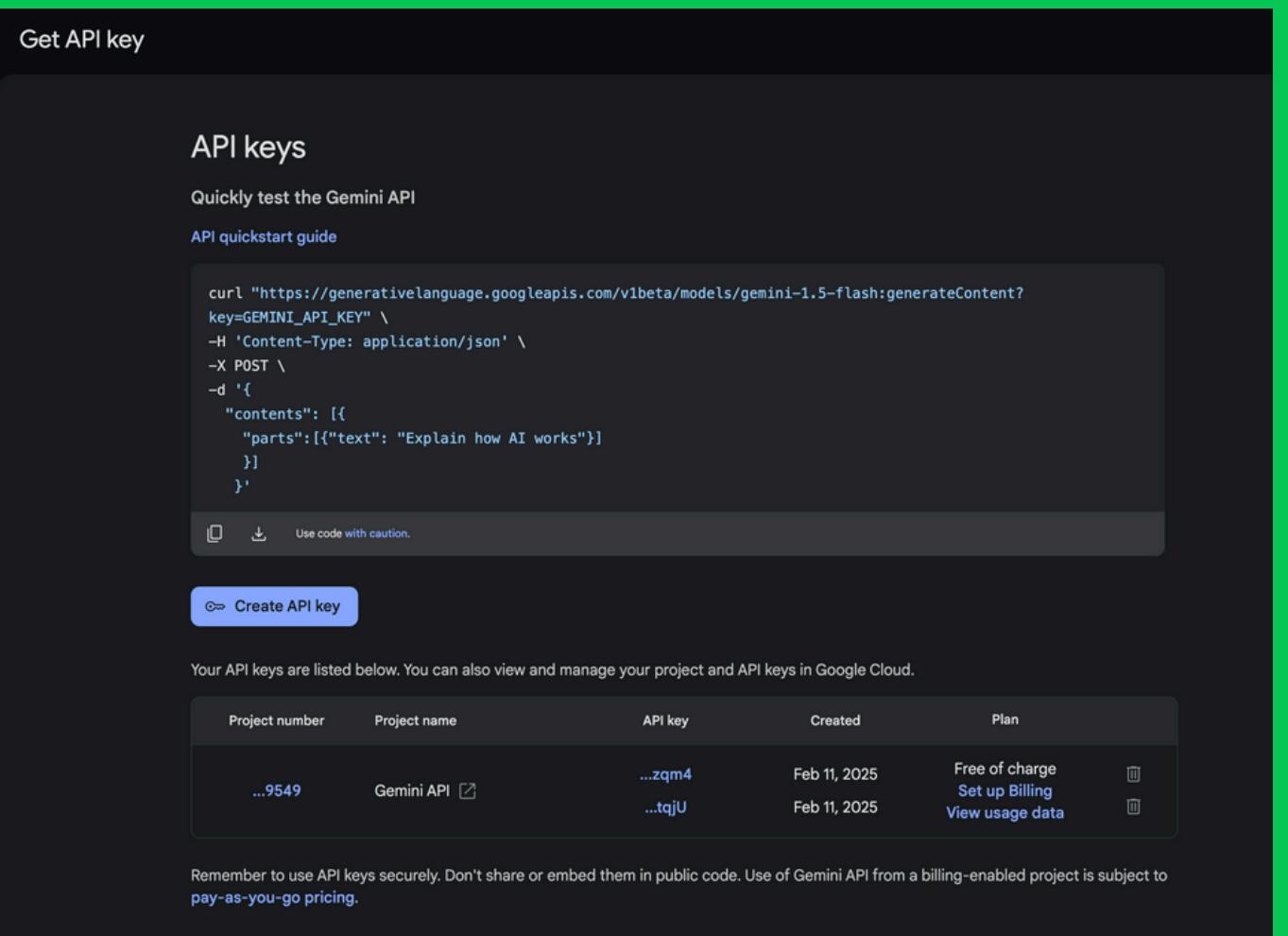
Copy the generated key and secret

`heroku config:set IMAGGA_KEY=[KEY_YOU_COPIED]`

`heroku config:set IMAGGA_SECRET=[SECRET_YOU_COPIED]`

Google Gemini

<https://aistudio.google.com/apikey>



Copy the generated key

heroku config:set GOOGLE_GEMINI_API_TOKEN=[KEY_YOU_COPIED]

THE END

Thank you for joining

Check out the bootcamp at
lewagon.com/tokyo



le wagon