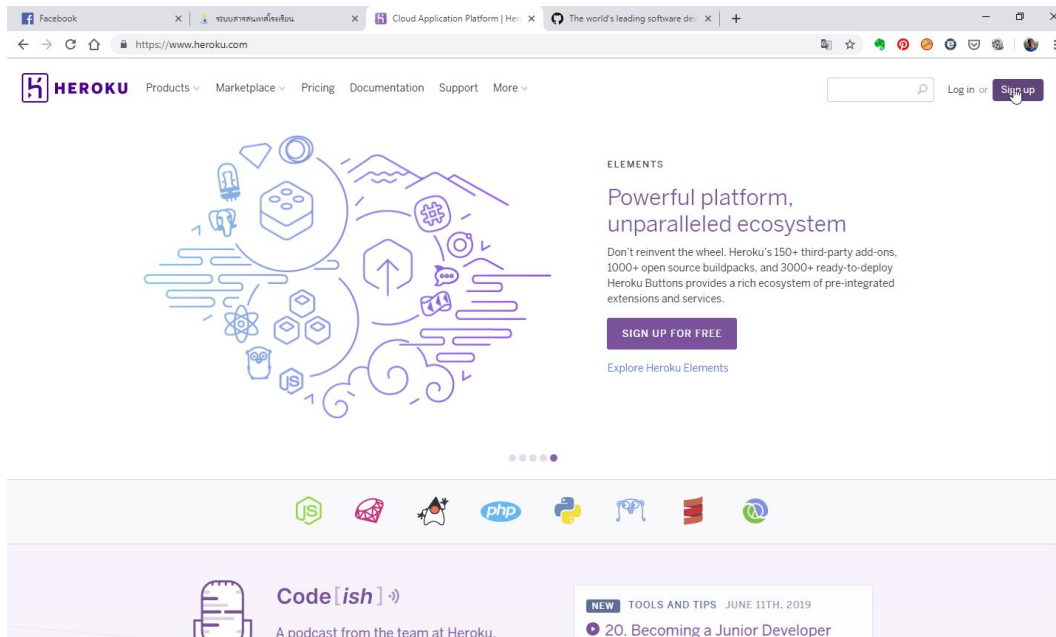


การลงทะเบียนเพื่อใช้งานเว็บ heroku.com

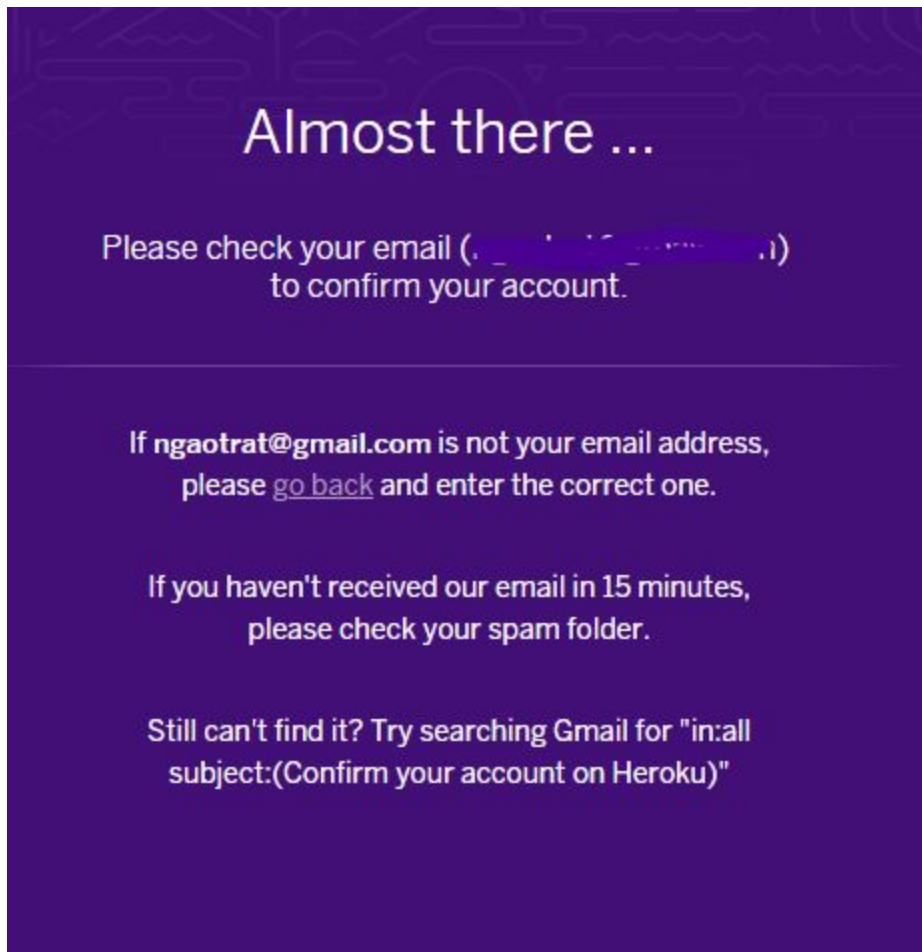
1. คลิกที่ Sign up



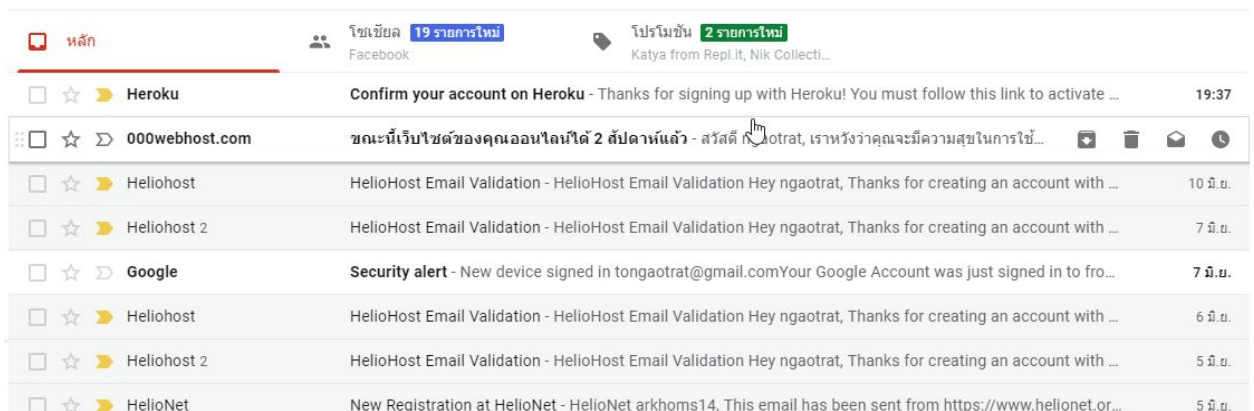
2. กรอกข้อมูล แล้วกดปุ่ม CREATE FREE ACCOUNT

A screenshot of the Heroku sign-up form. The background is a dark purple with a circuit-like pattern. The form is titled 'Sign up for free and experience Heroku today'. It has two main sections: 'Free account' on the left and a registration form on the right. The 'Free account' section describes the benefits of a free account and mentions 'Your app platform' and 'Deploy now'. The registration form on the right contains the following fields: 'First name' (Arkhom), 'Last name' (Suwanprasert), 'Email address' (arkhoms@streetrat.ac.th), 'Company name' (Streeprasertsin School), 'Role' (Student), 'Country' (Thailand), and 'Primary development language' (Python). At the bottom of the form is a checkbox for 'I'm not a robot' and a reCAPTCHA image.

3. ขึ้นข้อความแจ้งให้ตรวจสอบจดหมาย ใน e-mail ที่เราใช้สมัคร



4. คลิกที่จดหมายที่ได้รับจาก heroku.com



5. คลิกลิงค์ในจดหมาย



Thanks for signing up with Heroku! You must follow this link to activate your account:

<https://id.heroku.com/account/accept/6616049/6e317fba8a832a390bd7d1aa4edb0d0>

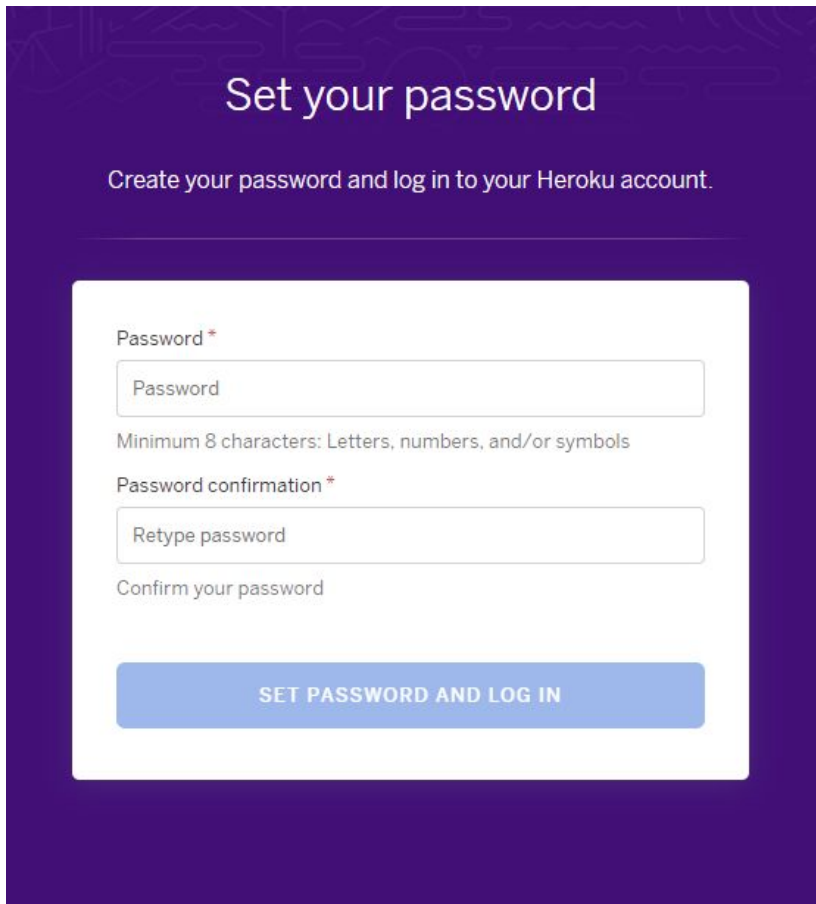
Have fun, and don't hesitate to contact us with your feedback.

The Heroku Team
<https://heroku.com>

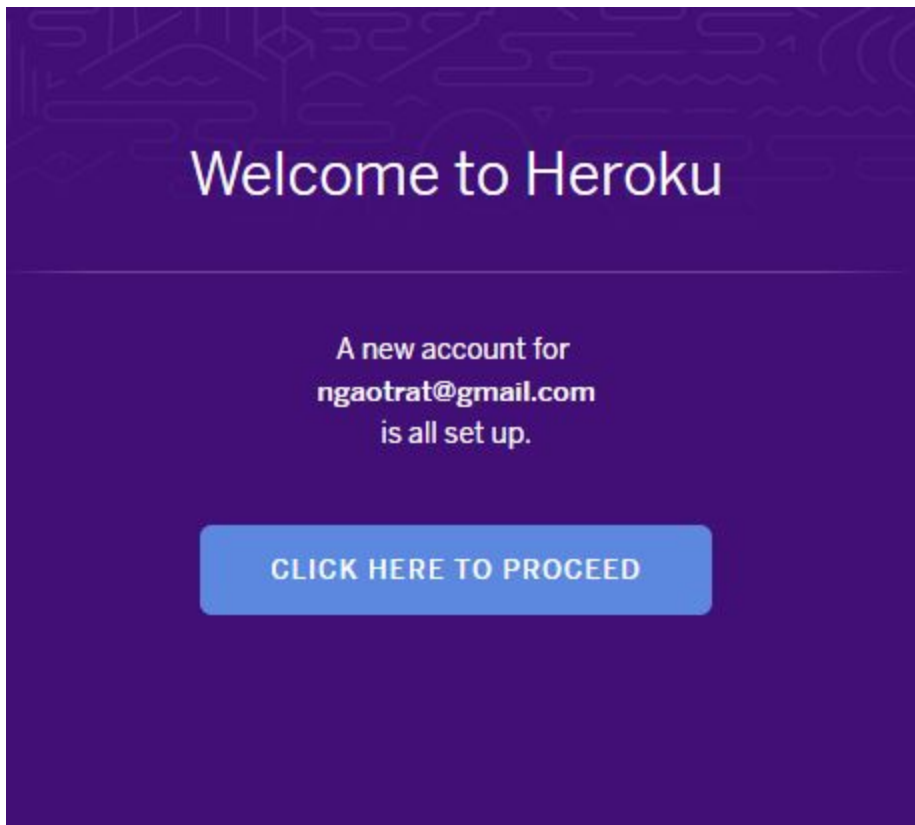
Heroku is the cloud platform for rapid deployment and scaling of web applications. Get up and running in minutes, then deploy instantly via Git.

To learn more about Heroku and all its features, check out the Dev Center:
<https://devcenter.heroku.com/articles/quickstart>

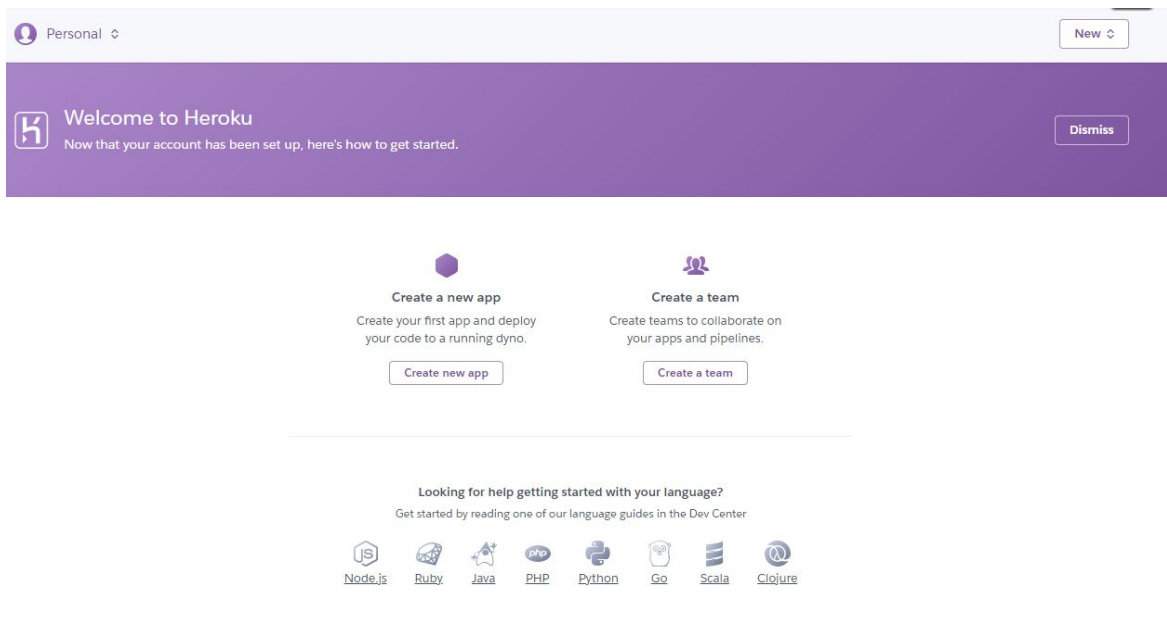
6. กำหนดรหัสผ่าน ต้องประกอบด้วย ตัวอักษร (อังกฤษ) ตัวเลข และสัญลักษณ์ โดยรวมกันไม่น้อยกว่า 8 ตัว



7. คลิกปุ่ม “CLICK HERE TO PROCEED”

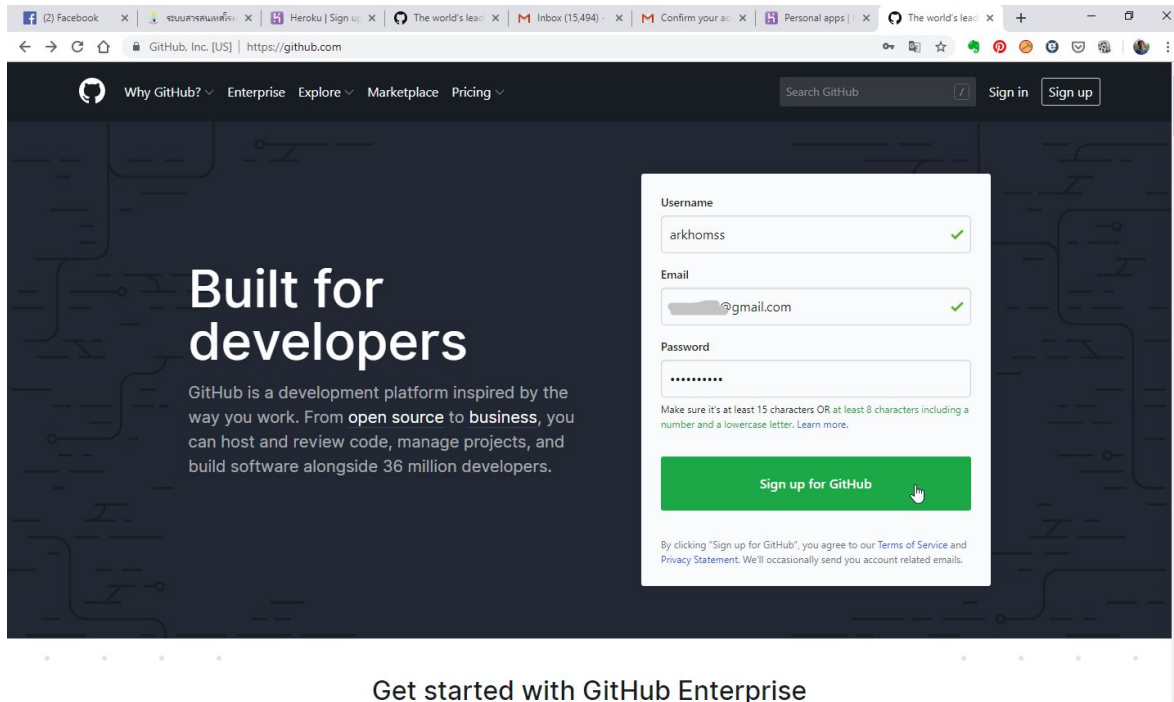


8. สำเร็จแล้ววว



การลงทะเบียนเพื่อใช้งานเว็บ github.com

1. ตั้ง Username กรอก e-mail และตั้งรหัสผ่าน (8-15 ตัว ประกอบด้วยตัวอักษรภาษาอังกฤษ และตัวเลข) แล้วกด Sign up for GitHub

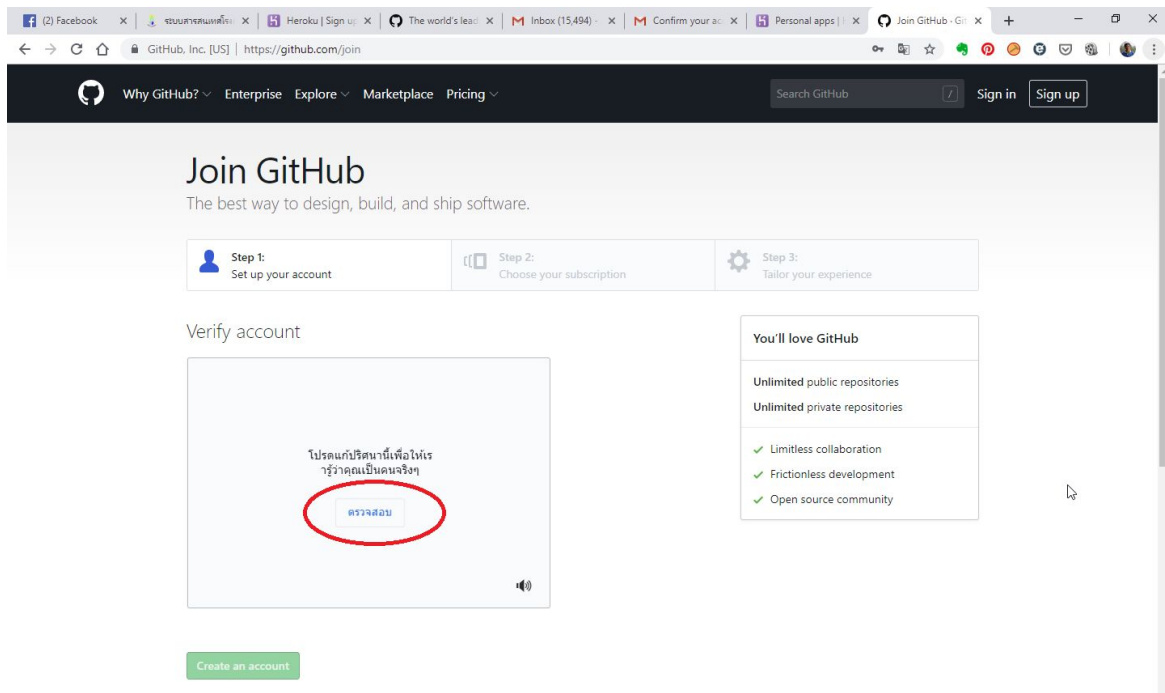


The screenshot shows the GitHub homepage with a sign-up modal open. The modal contains the following fields and text:

- Username:** arkhomss (with a green checkmark)
- Email:** [redacted]@gmail.com (with a green checkmark)
- Password:** [redacted] (with a green checkmark)
- Sign up for GitHub:** A large green button.
- Text below the button:** "By clicking 'Sign up for GitHub', you agree to our [Terms of Service](#) and [Privacy Statement](#). We'll occasionally send you account related emails."

Below the modal, the text "Get started with GitHub Enterprise" is visible.

2. แก้ปรีศนา โดยกด “ตรวจสอบ” แล้วทำตามภารกิจที่กำหนด




The screenshot shows the "Join GitHub" page with the following elements:


- Header:** "Join GitHub" and "The best way to design, build, and ship software."
- Steps:** Three steps are listed: "Step 1: Set up your account", "Step 2: Choose your subscription", and "Step 3: Tailor your experience".
- Verify account:** A section with a large button labeled "ตรวจสอบ" (Check) circled in red. Above the button is the text "โปรดแก้ปรีศนาเพื่อที่เราจะได้เป็นคุณจริงๆ" (Please correct the typo so we can be you for real).
- You'll love GitHub:** A list of benefits including "Unlimited public repositories", "Unlimited private repositories", "Limitless collaboration", "Frictionless development", and "Open source community".
- Create an account:** A green button at the bottom.


3. เลือกประเภทสมาชิก (ใช้แบบ Free)

Welcome to GitHub

You're a few steps away from building better software, @arkhomss.


 **Completed**
Set up your account

 **Step 2:**
Choose your subscription

 **Step 3:**
Personalize


Choose your subscription

With tools developers love and the world's largest open source community, there's no wrong choice.


Free
The basics of GitHub for every developer


\$0
per month


Includes:
∞ Unlimited public and private


Pro
Pro tools for developers with advanced requirements

\$7
per month
(view in THB)

4. แล้วกดปุ่ม Continue ด้านล่าง


repositories
✓ Issues and bug tracking
✓ Project management


repositories
∞ Unlimited collaborators
✓ Issues and bug tracking
✓ Project management
✓ Advanced tools and integrations

Are you a [student](#)? Get the best developer tools for free with [GitHub Student Developer Pack](#)

☐ **Help me set up an organization next**
Organizations are separate from personal accounts and are best suited for businesses who need to manage multiple repositories. [Learn more about organizations](#)

☐ **Send me updates on GitHub news, offers, and events**
Unsubscribe anytime in your email preferences. [Learn more](#)

Continue

5. ตอบคำถาม 2 ข้อ หรือคลิกที่ skip this step

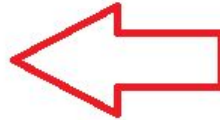
- ☐ Learning Git and GitHub
- ☐ Host a project (repository)
- ☐ Creating a website with GitHub Pages
- ☐ Collaborating with my team
- ☐ Finding a project to contribute to
- ☐ School work / School-related project
- ☐ The GitHub API
- ☐ I don't know yet
- ☐ Other (please specify)

What are you interested in?

What languages, frameworks, industries, or disciplines are you interested in?
e.g. fantasy-console, es6, xmake

Submit

[skip this step](#)



6. แจ้งให้เปิด e-mail

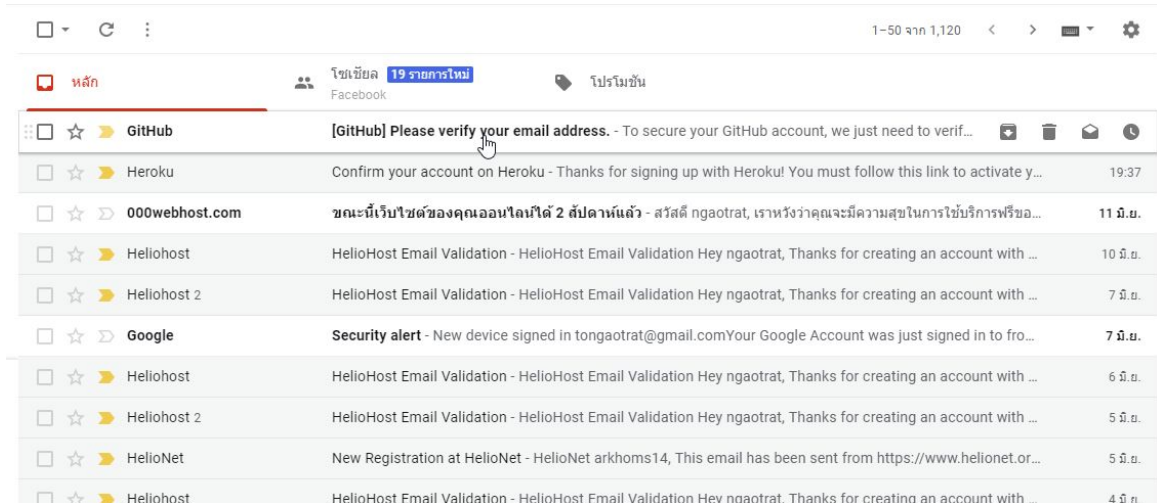


Please verify your email address

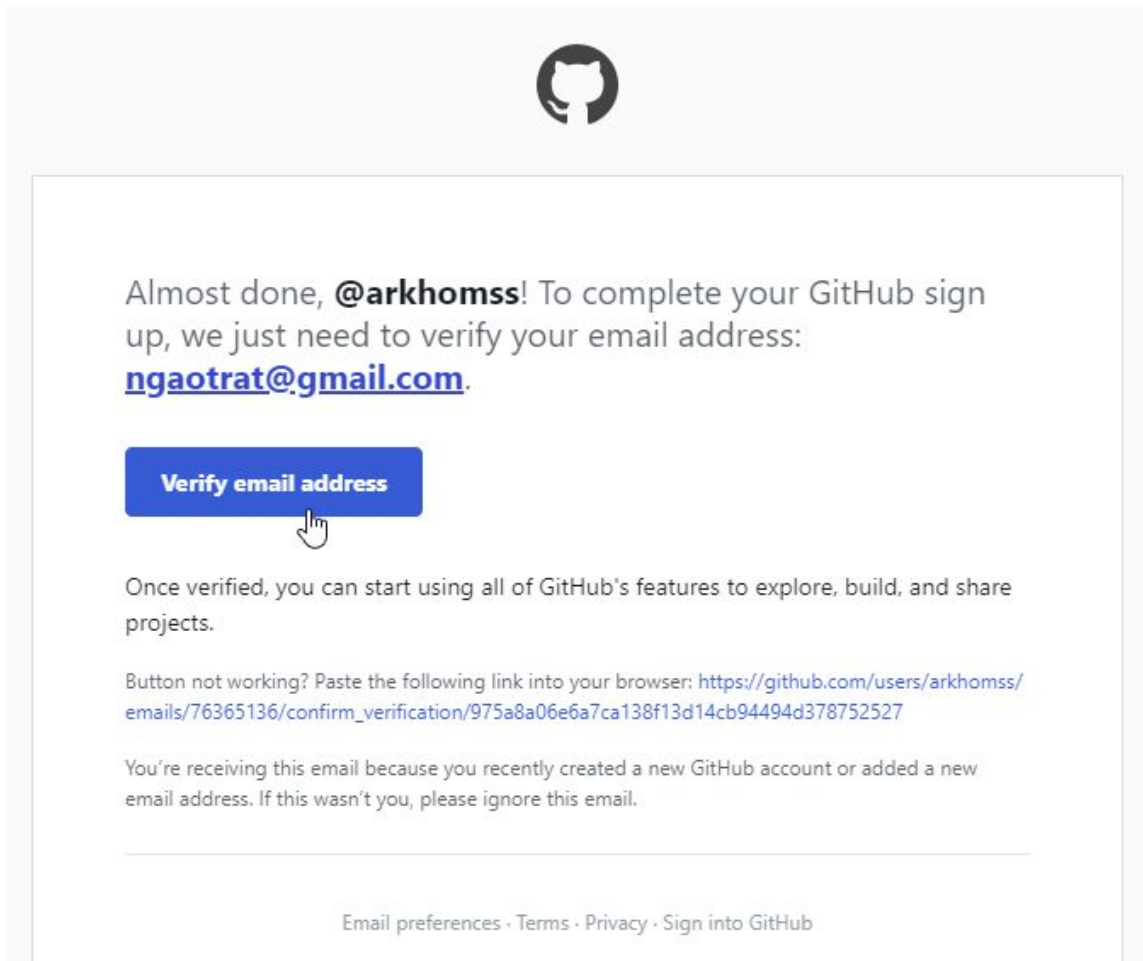
Before you can contribute on GitHub, we need you to verify your email address.
An email containing verification instructions was sent to **ngaotrat@gmail.com**.

Didn't get the email? [Resend verification email](#) or [change your email settings](#).

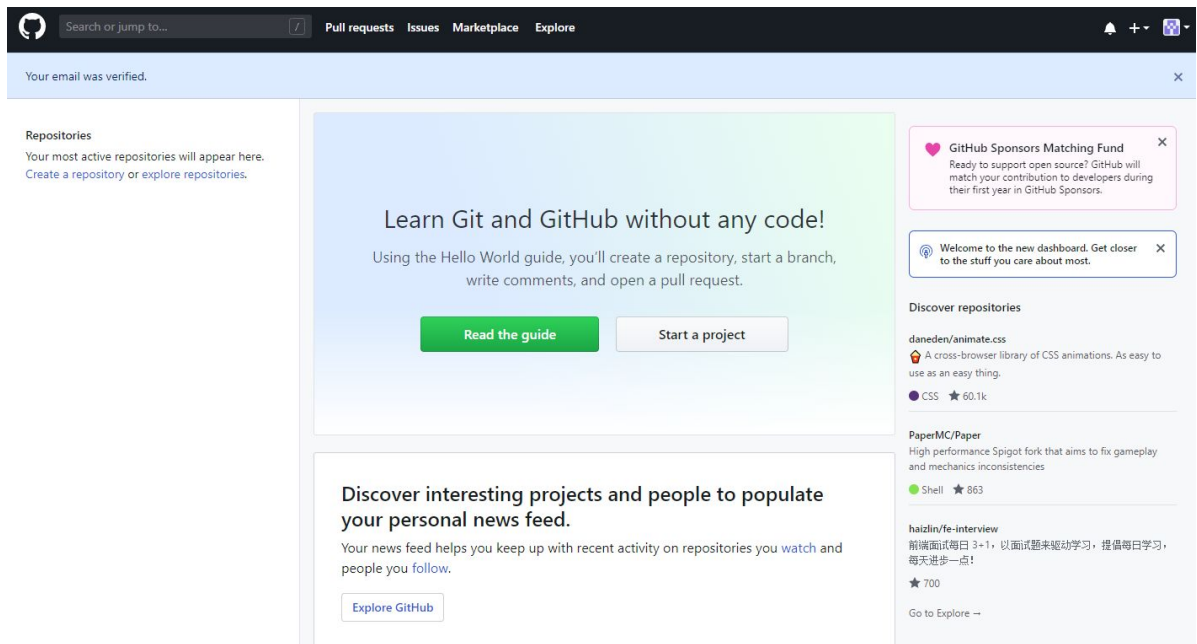
7. เข้าเว็บ e-mail (เช่น gmail.com) แล้วคลิกจดหมายที่ได้รับจาก github



8. คลิกที่ “Verify email address”



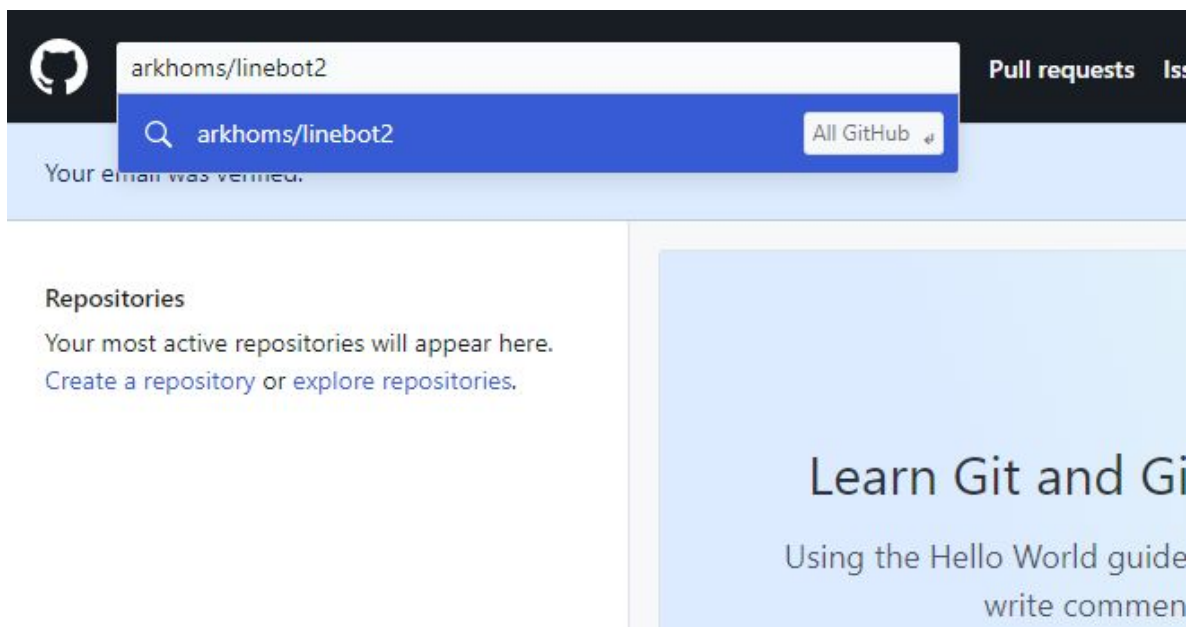
9. สำเร็จแล้วววว



การคัดลอกโค้ด (Repository) มาเป็นของเรา

แนวคิดของ GitHub คือการแบ่งปันโค้ด (ใน github จะให้สร้าง Repository เปรียบเสมือนโฟลเดอร์สำหรับเก็บไฟล์โปรแกรม) ซึ่งเราสามารถนำโค้ดโปรแกรมของผู้อื่นมาเป็นของเราแล้วใช้พัฒนาต่อได้ ดังนี้

1. ค้นหา Repository ที่เราต้องการ โดยพิมพ์ชื่อในช่องค้นหา (Search) ด้านซ้ายบน (ในที่นี้ให้พิมพ์ arkhoms/linebot2) แล้วกด enter



2. เมื่อพบแล้ว ให้คลิกที่ repository นั้น

The screenshot shows the GitHub search results for the repository 'arkhoms/lineBot2'. On the left, there is a sidebar with categories: Repositories (1), Code (0), Commits (0), Issues (0), Packages (0), Marketplace (0), Topics (0), Wikis (0), and Users (0). Below this is a 'Languages' section showing Python (1). The main content area is titled '1 repository result' and displays the repository 'arkhoms/lineBot2' with a red circle around the name and a hand cursor pointing to it. Below the name, it says 'Updated 9 hours ago'.

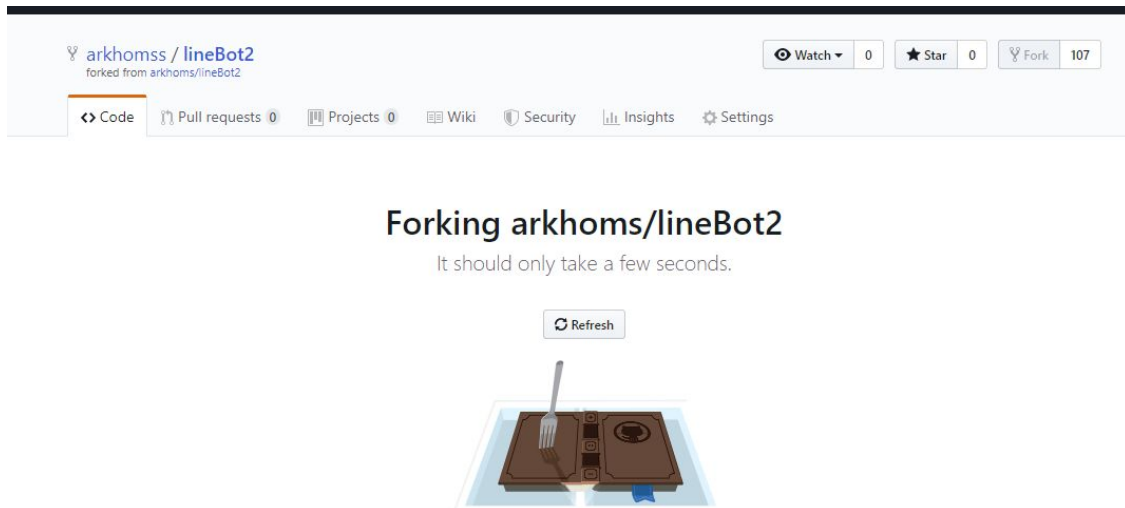
3. ก็จะเห็นไฟล์ต่างๆใน Repository นั้น เราสามารถคัดลอกทั้ง Repository มาเป็นของเราได้ โดยคลิกที่ Fork (เหมือนการใช้ส้อมจิ้มอาหารบนโต๊ะบุฟเฟ่)

The screenshot shows the GitHub repository page for 'arkhoms/lineBot2'. At the top, there are buttons for Watch (0), Star (0), and Fork (106). Below this, there are tabs for Code, Issues (0), Pull requests (0), Projects (0), Wiki, Security, and Insights. A red arrow points to the 'Fork' button. The repository has 98 commits, 1 branch, 0 releases, and 1 contributor. Below this, there is a table of files and their commit history:

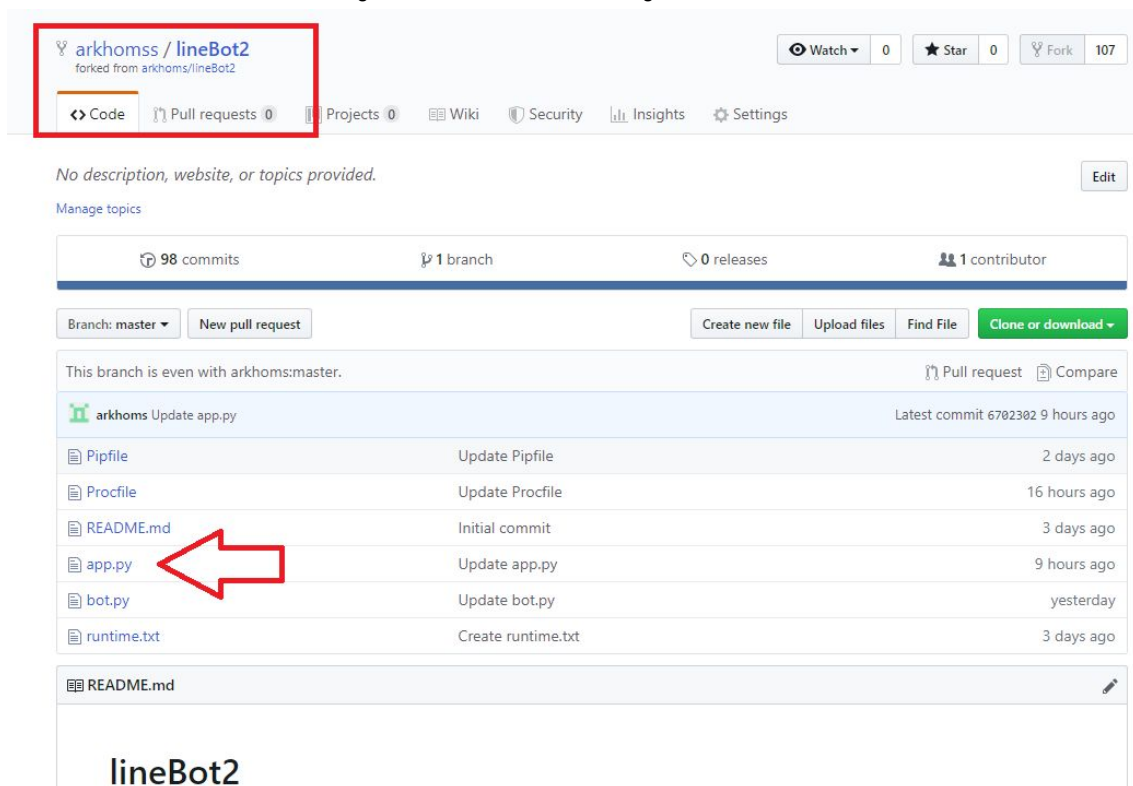
File	Commit	Time
Pipfile	Update Pipfile	2 days ago
Procfile	Update Procfile	16 hours ago
README.md	Initial commit	3 days ago
app.py	Update app.py	9 hours ago
bot.py	Update bot.py	yesterday
runtime.txt	Create runtime.txt	3 days ago

Below the table, there is a section for the README.md file, which contains the text 'lineBot2'.

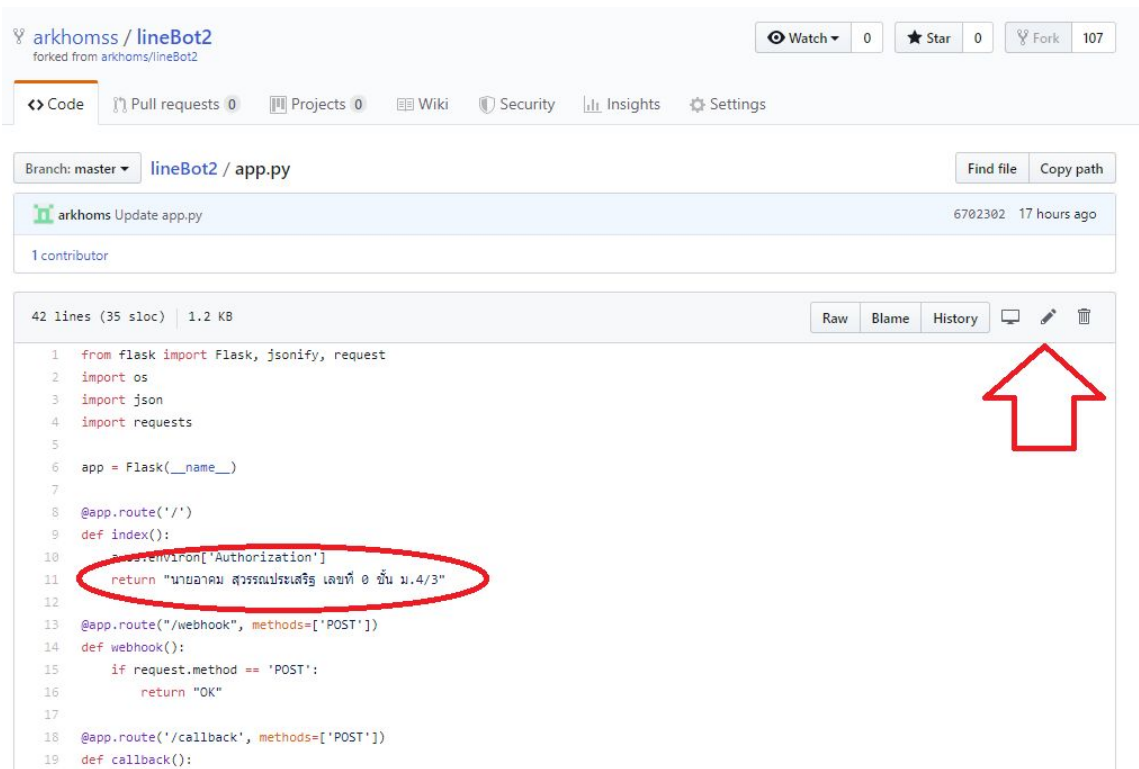
4. github จะทำการคัดลอกสักครู่



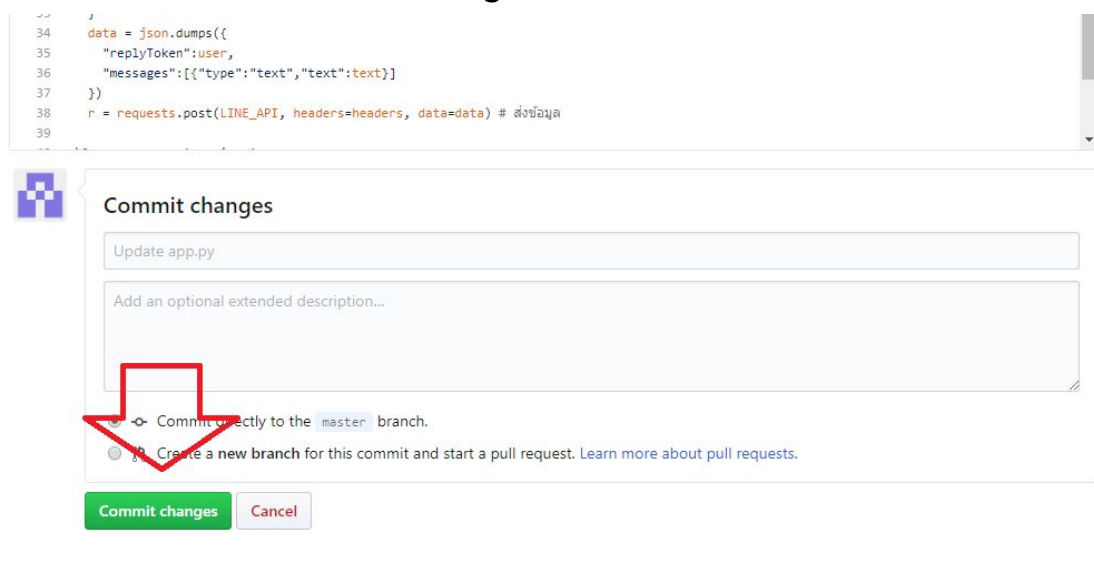
5. แล้วจะแสดงไฟล์ใน Repository ของเรา ซึ่งจะเหมือนกับต้นฉบับ แต่จะต่างกันตรงซ้ายบน ซึ่งแสดงชื่อผู้ใช้ของเรา และชื่อผู้ใช้ที่เรา forked มา



6. ทดลองแก้ไขโค้ดของเรา (ที่ fork มา) ในที่นี้ให้คลิกที่ไฟล์ app.py แล้วคลิกที่ไอคอนรูปดินสอ เพื่อใช้โหมดแก้ไข



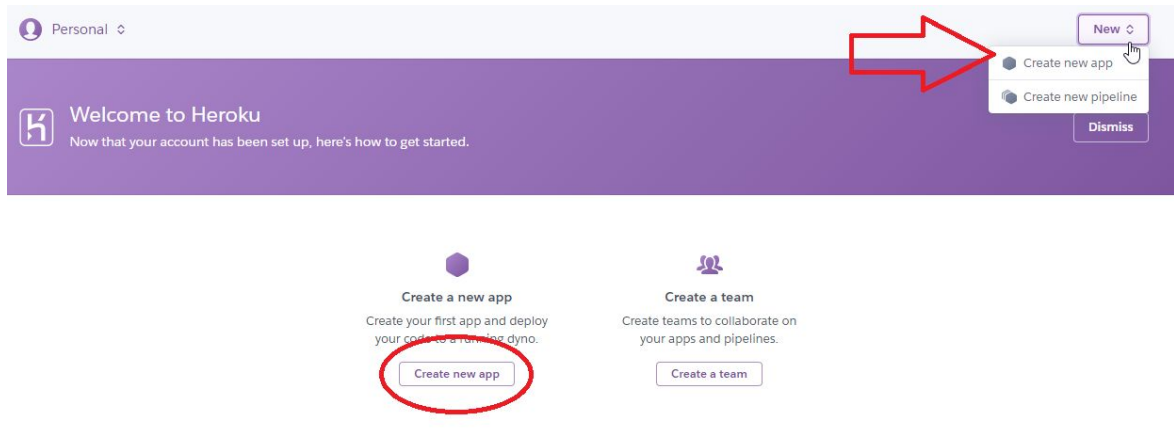
7. แก้ชื่อ เลขที่ ชั้น (บรรทัดที่ 11) ให้เป็นชื่อ เลขที่ ชั้น ของเรา จากนั้นเลื่อนลงด้านล่าง แล้วคลิกที่ Commit changes



8. จะกลับมาที่หน้าแสดงโค้ด

การสร้าง webapp ด้วย heroku.com

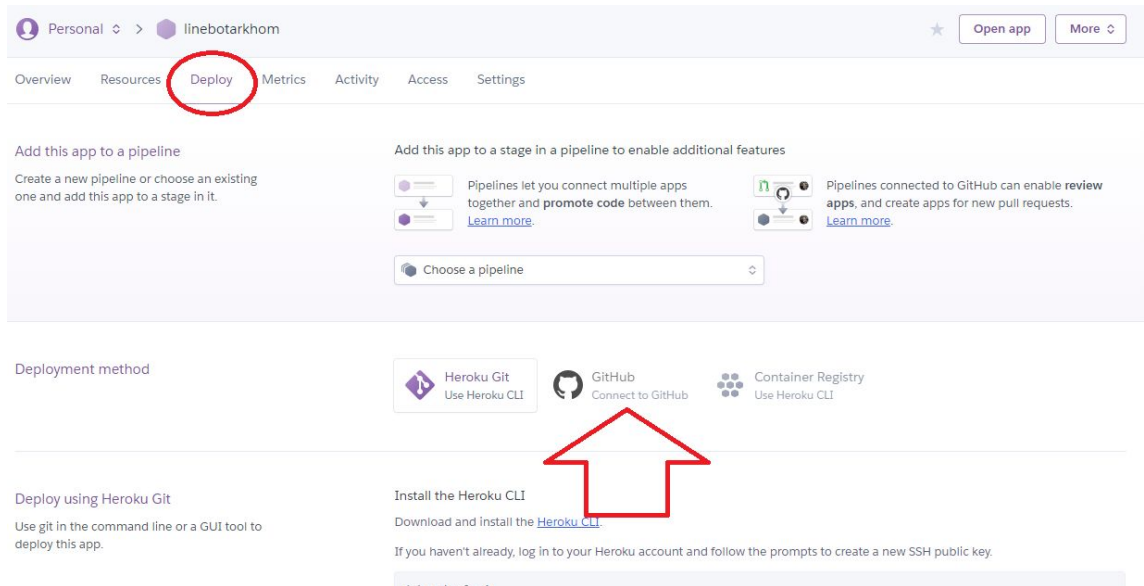
1. เข้าเว็บ heroku.com และล็อกอินให้เรียบร้อย จากนั้นคลิกที่ New -> Create new app



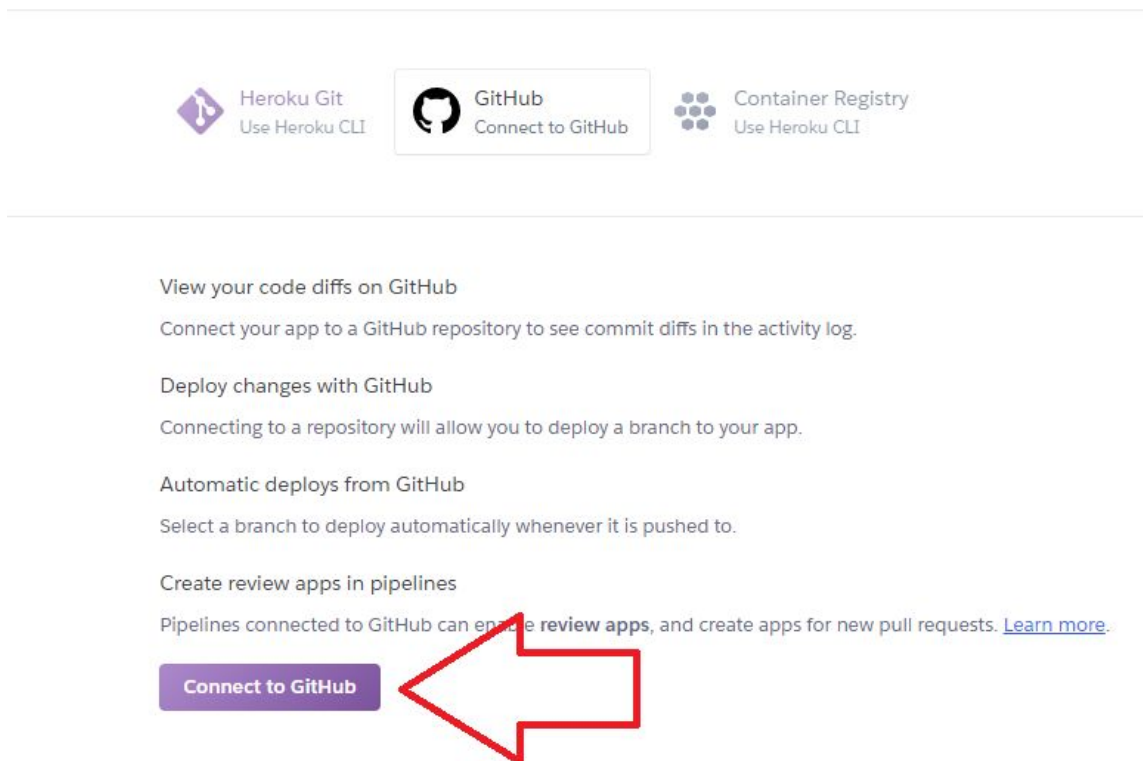
2. ตั้งชื่อแอป ด้วยคำภาษาอังกฤษ (รวมตัวเลข ไม่เว้นวรรค) แล้วคลิกที่ Create app (ส่วนอื่นไม่ต้องแก้ไข)

A screenshot of the 'Create New App' form on Heroku. The form has a title 'Create New App' at the top. Below it, there is a text input field for 'App name' with the placeholder text 'app-name'. A red arrow points to this field. Below the 'App name' field is a 'Choose a region' section with a dropdown menu showing 'United States'. At the bottom of the form, there is a 'Create app' button.

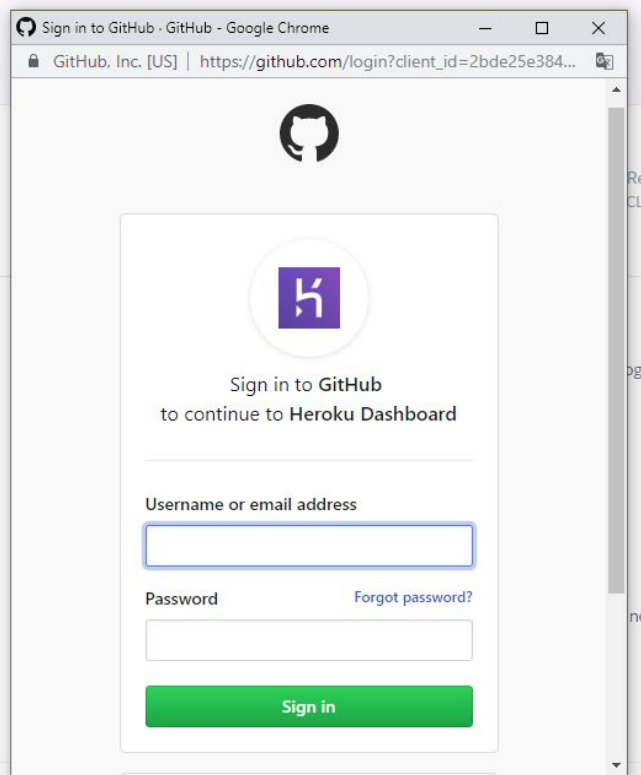
3. ก็จะมาถึงขั้นตอนการนำโค้ด (โปรแกรม) มาใช้งาน (หากไม่ขึ้นหน้านี้ ให้คลิก Deploy ตามวงกลมในภาพก่อน) ซึ่งสามารถทำได้ 3 วิธี ในที่นี้เราจะใช้วิธีเชื่อมโยง (ใช้โค้ดใน) กับ GitHub ดังนั้นให้คลิกตามที่ลูกศรชี้



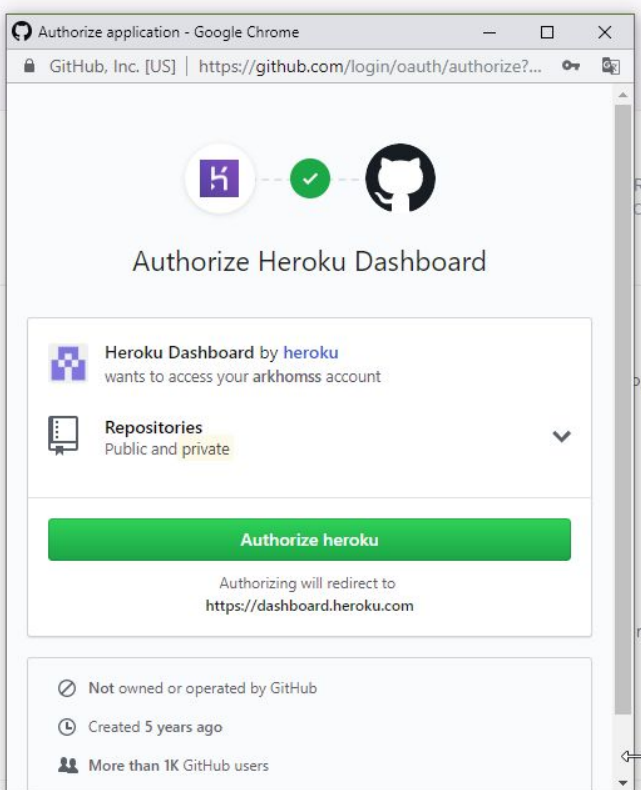
4. คลิกที่ Connect to GitHub



5. หากยังไม่ได้ล็อกอินเข้า GitHub ไว้ก่อน ก็จะขึ้นหน้าต่างให้เรากรอก Username หรือ อีเมล และรหัสผ่าน ที่เราสมัครไว้กับ GitHub แล้วกด Sign in



ถ้าล็อกอินแล้ว ก็ให้คลิกที่ Authorize heroku ได้เลย



6. ในวงกลมจะแสดงชื่อผู้ใช้ (Username) ที่เราสมัครใน GitHub ให้คลิกที่ปุ่ม Search แล้วกด Connect หลังรายการ Repository ที่ต้องการ (ในที่นี้มีเพียง lineBot2)

Heroku Git Use Heroku CLI

GitHub Connect to GitHub

Container Registry Use Heroku CLI

Search for repository to connect to

arkhomss repo-name Search

Missing a GitHub organization? Ensure Heroku Dashboard has team access.

arkhomss/lineBot2 Connect

7. เลื่อนลงด้านล่าง แล้วกด Enable Automatic Deploys และ Deploy Branch ตามลำดับ

Connected to arkhomss/lineBot2 by arkhomss Disconnect...

Releases in the [activity feed](#) link to GitHub to view commit diffs

Enable automatic deploys from GitHub

Every push to the branch you specify here will deploy a new version of this app. **Deploys happen automatically:** be sure that this branch is always in a deployable state and any tests have passed before you push. [Learn more.](#)

Choose a branch to deploy

master

☐ Wait for CI to pass before deploy

Only enable this option if you have a Continuous Integration service configured on your repo.

Enable Automatic Deploys

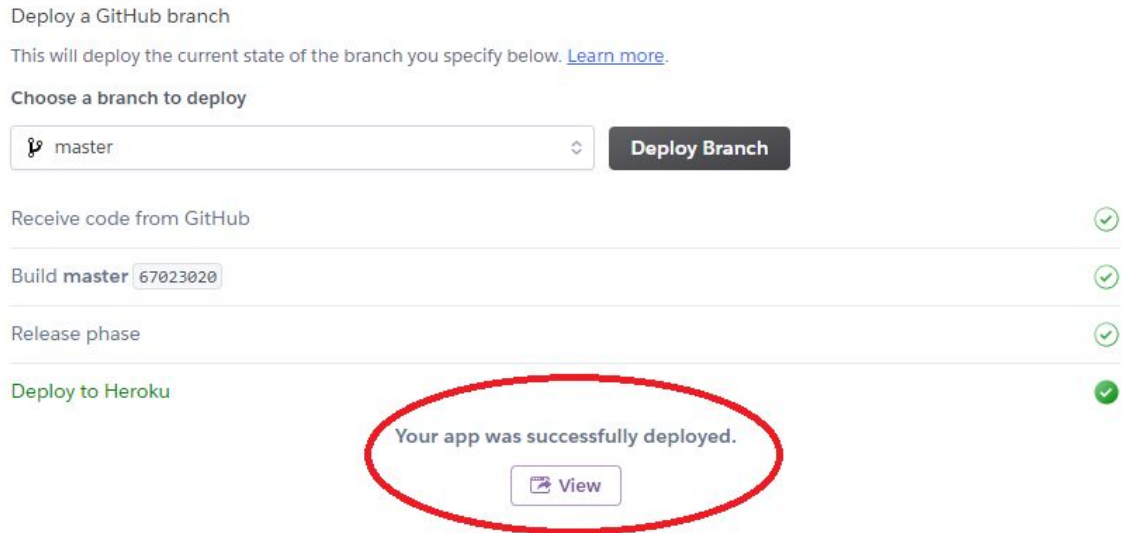
Deploy a GitHub branch

This will deploy the current state of the branch you specify below. [Learn more.](#)

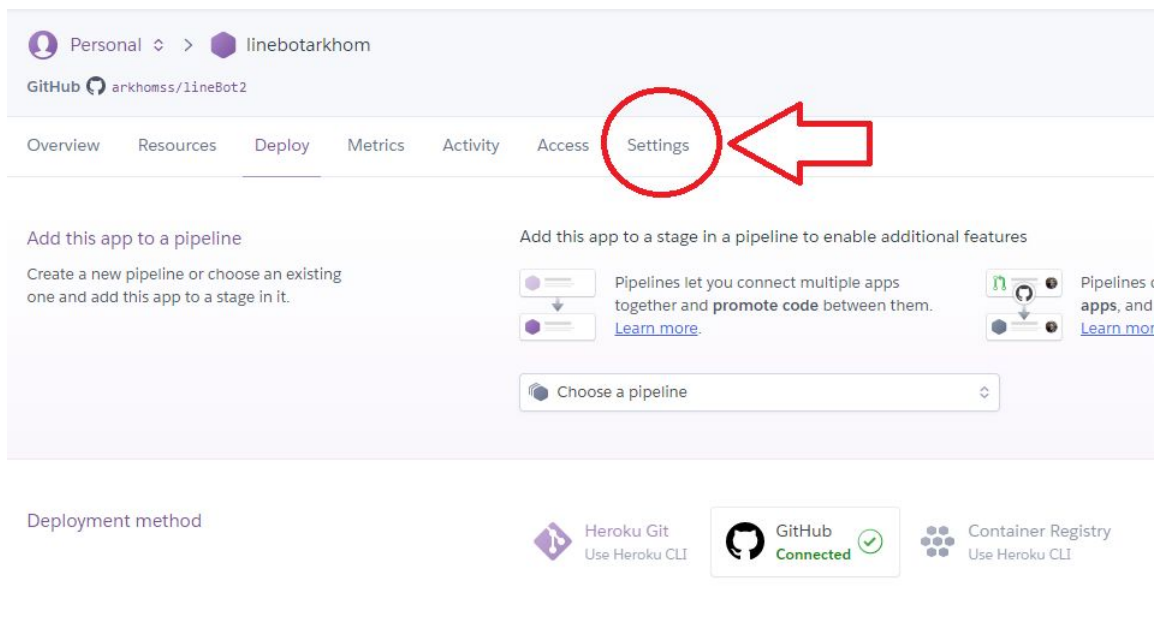
Choose a branch to deploy

master Deploy Branch

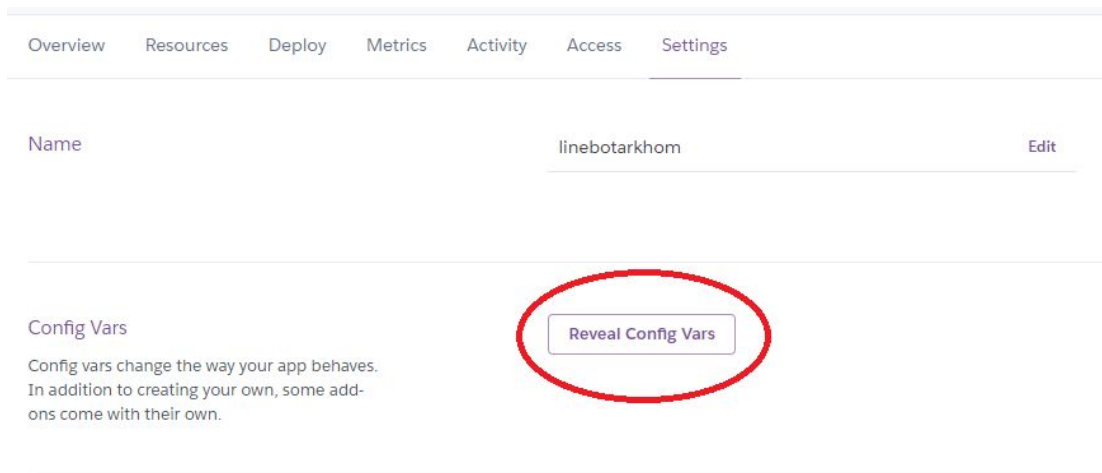
8. รอสักครู่ จนกว่าจะขึ้น Your app was successfully deployed ดังภาพ



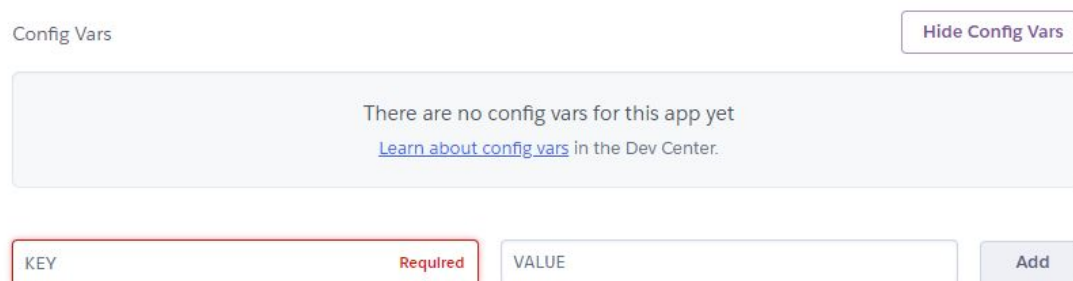
9. เลื่อนขึ้นด้านบน แล้วคลิกที่ Setting



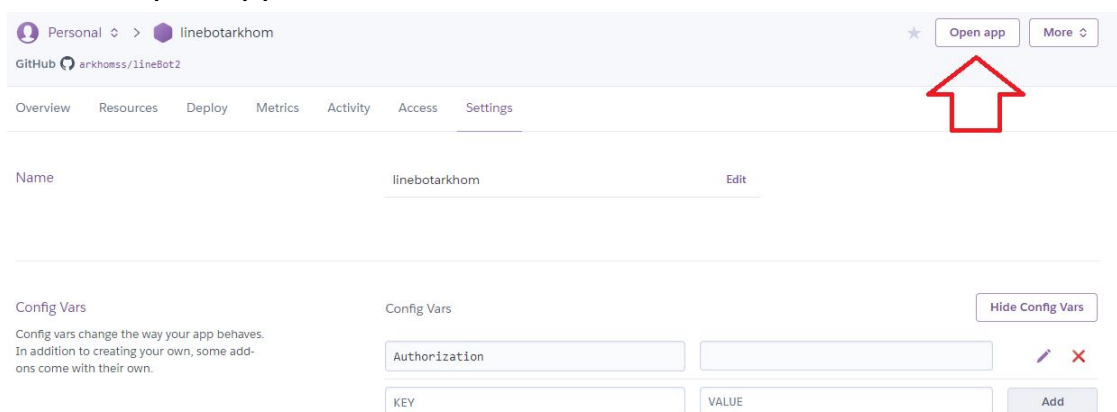
10. คลิกที่ Reveal Config Vars



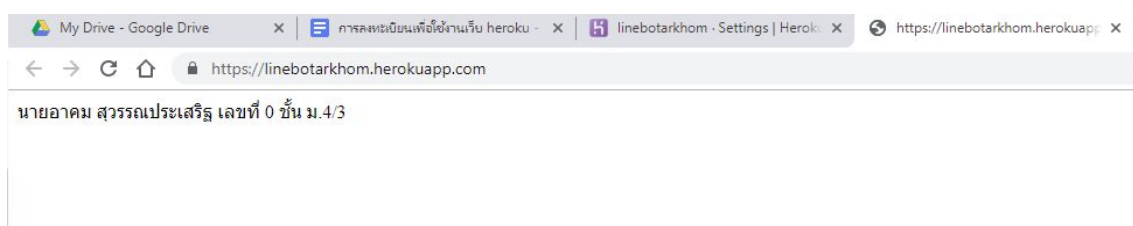
11. ตั้งชื่อตัวแปรระบบ (ในที่นี้ให้ใส่ Authorization) ในช่อง Key แล้วกด Add



12. กดที่ Open app

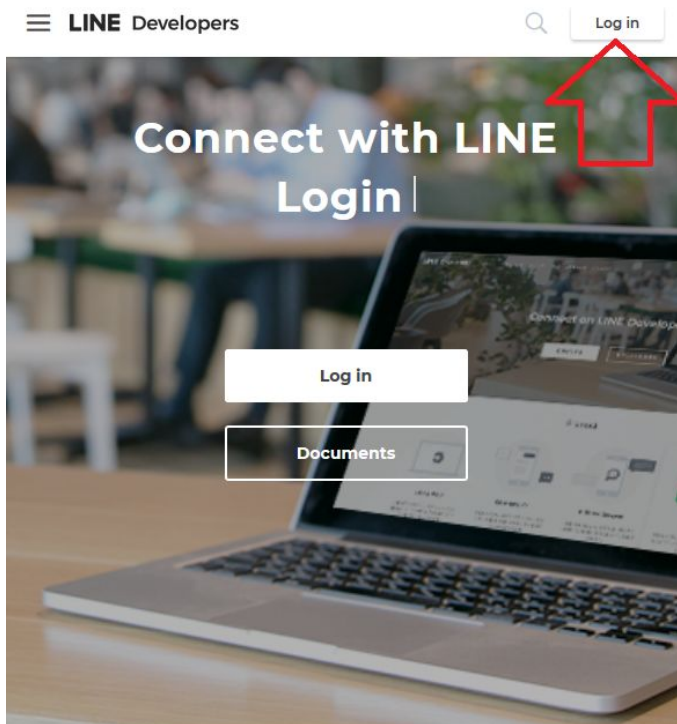


13. ก็จะแสดงหน้าเว็บแอพของเราขึ้นมา (ดังภาพ)



สร้าง Line messaging api channel

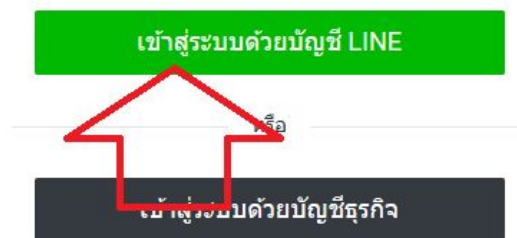
1. เข้าเว็บ developers.line.me แล้วกดที่ Log in



2. กด “เข้าสู่ระบบด้วยบัญชี LINE” (ต้องลงทะเบียนอีเมลก่อน)

[กลับไปบริการ](#)

LINE Business ID



สร้างบัญชี

การใช้ LINE Business ID จะเป็นไปตามที่กำหนดในข้อกำหนดการใช้บริการ
เมื่อคุณเข้าสู่ระบบหมายความว่าคุณยอมรับข้อกำหนดดังกล่าวแล้ว

② เกี่ยวกับ LINE Business ID

ภาษาไทย

[ช่วยเหลือ](#) [ข้อกำหนดการใช้บริการ](#) © LINE Corporation

3. ป้อนอีเมล และรหัสผ่าน แล้วกด “เข้าสู่ระบบ”



The image shows the LINE login interface. At the top is the green LINE logo. Below it are two input fields: the first is labeled 'อีเมล' (Email) and the second is labeled 'รหัสผ่าน' (Password). Below these fields is a large blue button with the text 'เข้าสู่ระบบ' (Log In) and a red prohibition sign (a circle with a diagonal line) to its right. Below the button is a link that says 'เข้าสู่ระบบด้วยวิธีอื่น' (Log in with other methods). At the bottom is a link that says 'ลืมอีเมลหรือรหัสผ่าน?' (Forgot email or password?).

4. สร้าง Provider (หากยังไม่เคยสร้าง) โดยกดที่เครื่องหมาย +



Welcome to LINE Developers Console !

Let's develop an app that connects people with people using your development technology and LINE Platform!

A provider is a service provider (company / individual), and we begin by creating a provider.

STEP 1

5. ตั้งชื่อ Provider (ห้องเก็บ Channel-lineBot) ตามต้องการ แล้วกด Confirm

Create new provider

Enter provider information Confirm Done

Enter name of provider
The provider is the entity (individual or company) that offers the app.

Provider name

Enter name of provider
Max: 100 characters

Confirm

6. ยืนยัน โดยกดที่ Create

Create new provider

Enter provider information Confirm Done

Enter name of provider
The provider is the entity (individual or company) that offers the app.

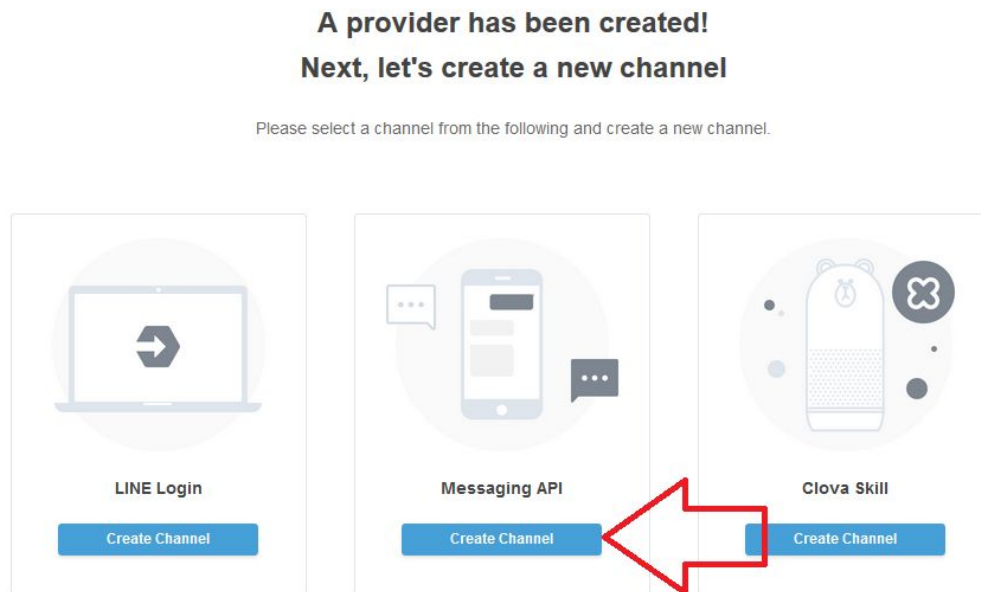
Provider name

โรงเรียน

Create

Edit

7. สร้าง Messaging API Channel โดยกด ดังรูป




8. ป้อนข้อมูล Channel (LineBot) โดยในเบื้องต้นต้องใส่ App name (ชื่อ LineBot), App description (อธิบายเพิ่มเติม), Category, Subcategory และ Email address แล้วกด Confirm

Enter channel information Confirm

Enter information for the Messaging API

Selected Provider โรงเรียน

App icon optional

 Register

Under 3MB; JPEG/PNG/GIF/BMP

App name

Enter app name

Max: 20 characters

Note: The app name cannot be changed for seven days.

9. กด “ยอมรับ” ข้อตกลง

ข้อตกลงเกี่ยวกับการใช้ข้อมูล

LINE คอร์ปอเรชั่น (“LINE”) จะมีการใช้ข้อมูลประเภทต่างๆ จากบัญชีของธุรกิจ ซึ่งรวมถึงแต่ไม่จำกัดเฉพาะบัญชีทางการ LINE และบัญชี LINE@ (เรียกรวมกันว่า “บัญชีทางการ”) เพื่อพัฒนาและปรับปรุงการให้บริการ กรุณาตรวจสอบและยอมรับข้อตกลงการใช้ข้อมูลก่อนใช้บัญชีทางการ

■ ข้อมูลที่จะเก็บและนำไปใช้

- เนื้อหาต่างๆ ที่มีการส่ง ทั้งที่ส่งถึงผู้ใช้และส่งจากผู้ใช้ เช่น ข้อความ ข้อมูล URL รูปภาพ วิดีโอ สติกเกอร์ และเอฟเฟกต์ เป็นต้น
- รูปแบบ ตัวเลข เวลาในการส่งหรือโทร คู่สนทนา และอื่นๆ (“รูปแบบและอื่นๆ”) ของเนื้อหาทั้งหมดที่ส่งถึงหรือส่งจากผู้ใช้ รวมถึงรูปแบบและอื่นๆ ของเนื้อหาที่มีการจัดการใน VoIP (การโทรทางอินเทอร์เน็ตและการโทรทางวิดีโอ) และฟังก์ชันอื่นๆ
- ข้อมูลที่ใช้สำหรับบัญชีทางการ เวลาที่ใช้ในแต่ละฟังก์ชัน สถานะว่าเนื้อหาถูกอ่านแล้วหรือไม่ รวมถึงการแตะหรือคลิก URL และอื่นๆ (รวมถึงแต่ไม่จำกัดเฉพาะฟังก์ชันที่เชื่อมโยงไปยังแหล่งข้อมูล) ประวัติการใช้งานบริการ เช่น ประวัติการดูเว็บเบราว์เซอร์ผ่าน LINE และระยะเวลาที่ดู ตลอดจนข้อมูลอื่นๆ ที่ระบุไว้ในนโยบายความเป็นส่วนตัว

■ วัตถุประสงค์ในการเก็บข้อมูล การใช้ และการส่งต่อไปกับบุคคลภายนอก

ข้อมูลข้างต้นจะถูกนำไปใช้สำหรับ (1) การป้องกันการใช้งานโดยไม่ได้รับอนุญาต, (2) การให้บริการตลอดจนการพัฒนาและปรับปรุงบริการต่างๆ ของ LINE และ (3) การแสดงโฆษณาต่างๆ

นอกจากนี้ข้อมูลดังกล่าวอาจมีการส่งต่อไปกับบริษัทต่างๆ ในเครือของ LINE, บริษัทผู้ให้บริการ และบริษัทพันธมิตรของ LINE

ข้อตกลงดังกล่าวมีไว้สำหรับผู้มีอำนาจในการใช้งานบัญชีทางการเท่านั้น หากเป็นบุคคลอื่น จำเป็นต้องได้รับความเห็นชอบจากผู้มีอำนาจ มิเช่นนั้น LINE อาจระงับการใช้งานบัญชีทางการหากได้รับแจ้งจากผู้มีอำนาจว่าไม่ได้รับการเห็นชอบ โดย LINE จะไม่มีส่วนเกี่ยวข้องหรือรับผิดชอบกับผลที่เกิดขึ้นในภายหลัง

ยอมรับ

10. เลือกยอมรับเงื่อนไขการใช้งาน แล้วกด Create

Privacy Policy URL

Terms Of Use URL

☒ LINE Official Account Terms of Use [🔗](#) I have read and agree to the Terms of Use.

☒ LINE Official API Terms of Use: [🔗](#) I have read and agree to the Terms of Use.

Edit Create

11. สำเร็จ

Create new channel

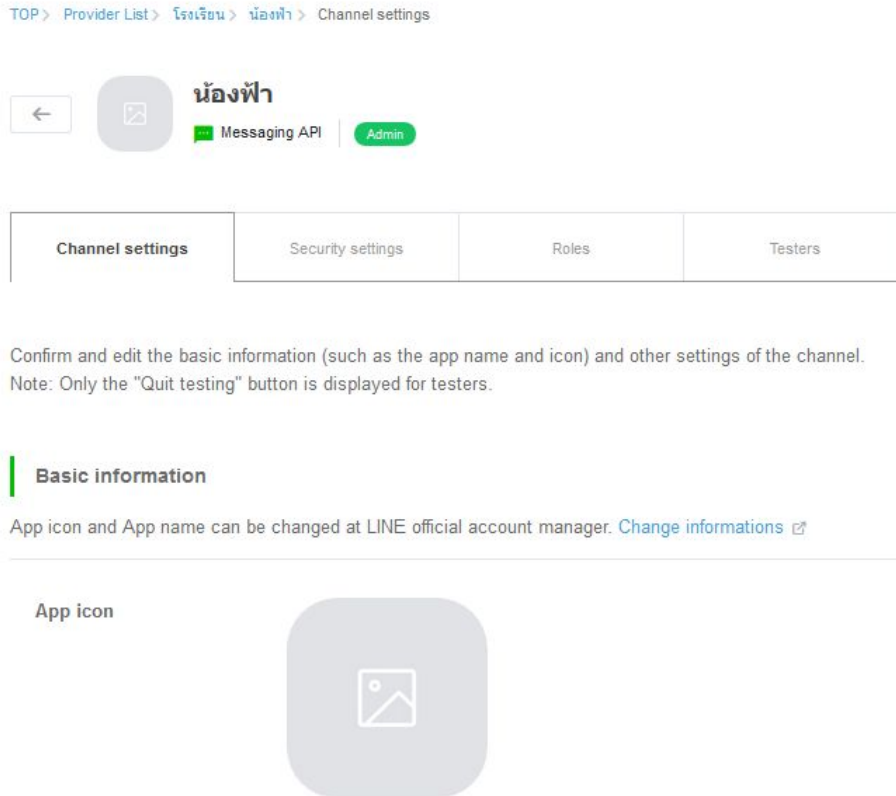
New Admin

น้องฟ้า

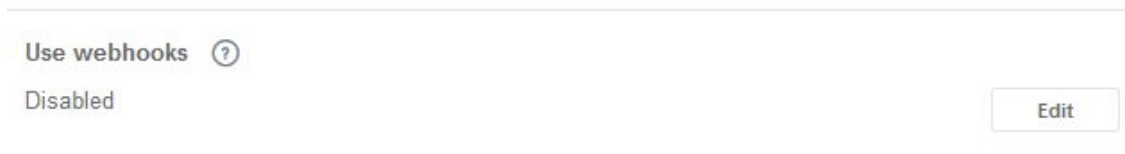
Messaging API

การเชื่อม Line messaging api Channel เข้ากับ Heroku webapp

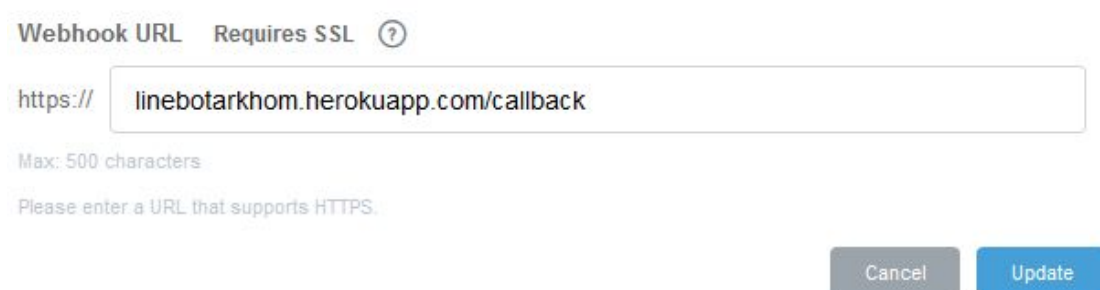
1. กดที่ Channel ที่ต้องการเชื่อม (ในที่นี้คือ “น้องฟ้า”)



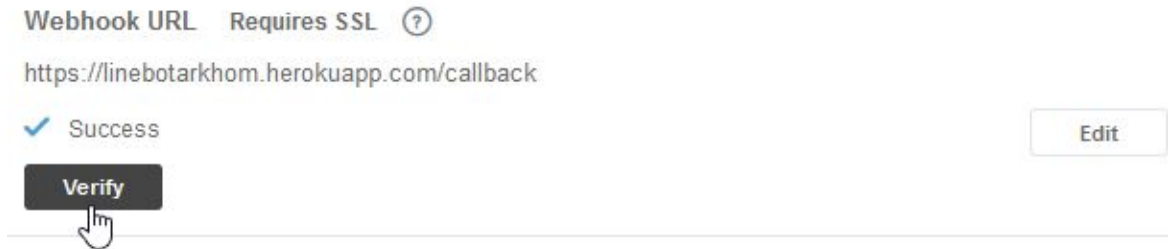
2. เลื่อนลงมาที่หัวข้อ Use webhooks กด Edit ปรับให้เป็น Enabled แล้วกด Update



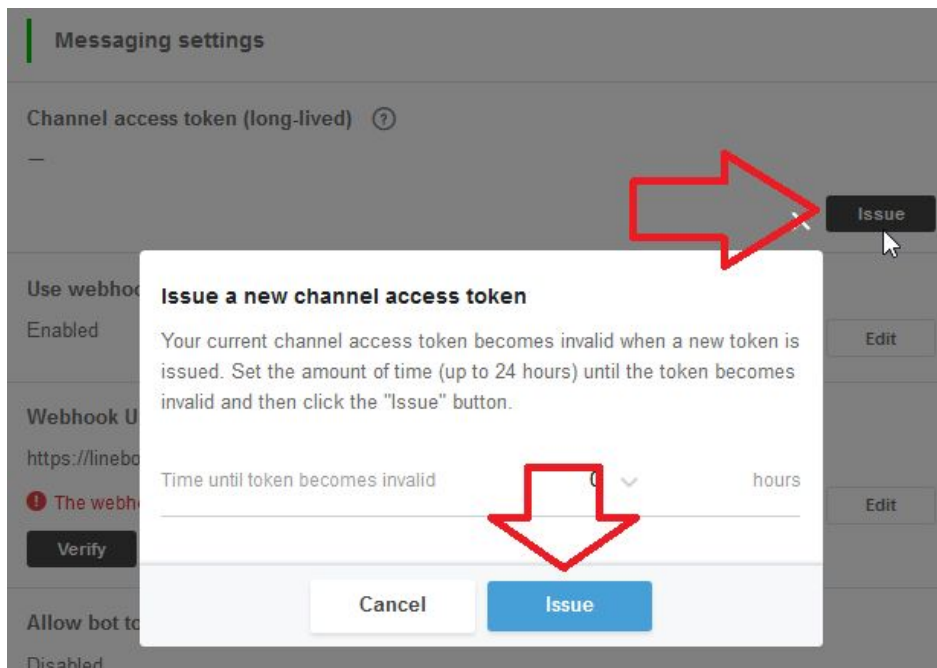
3. และหัวข้อ Webhook URL กด Edit แล้วป้อน URL ของ Heroku webapp ตามด้วย /callback เช่น linebotarkhom.herokuapp.com/callback แล้วกด Update



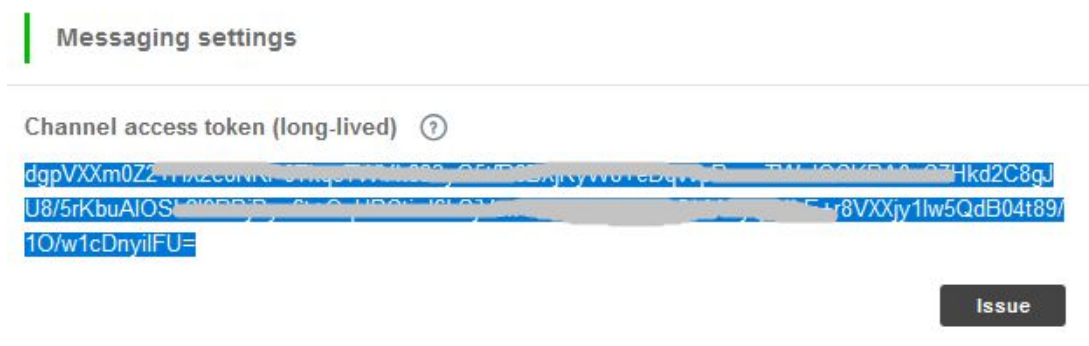
4. กดปุ่ม Verify เพื่อตรวจสอบการเชื่อมต่อ ซึ่งต้องแสดง Success กลับมา



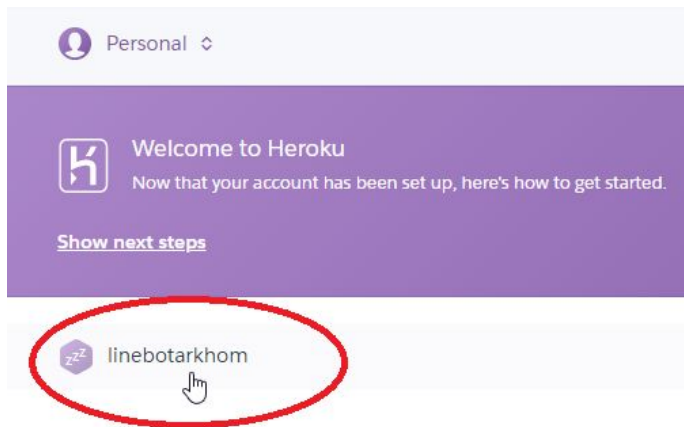
5. ที่หัวข้อ Channel access token(long-lived) ให้กด issue เมื่อมีหน้าต่างปรากฏมาก็ให้กด Issue อีกครั้ง



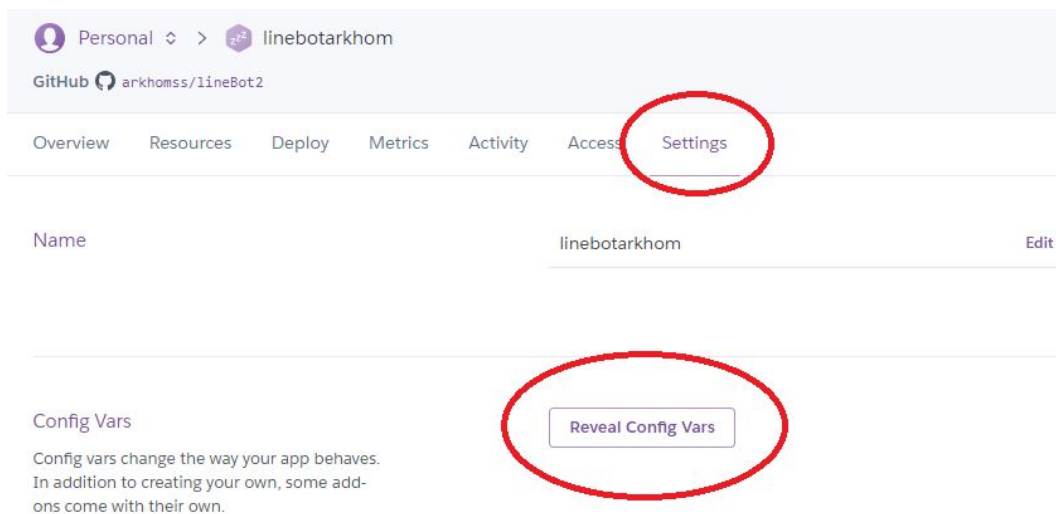
6. คัดลอกข้อความที่ปรากฏขึ้นมาทั้งหมด



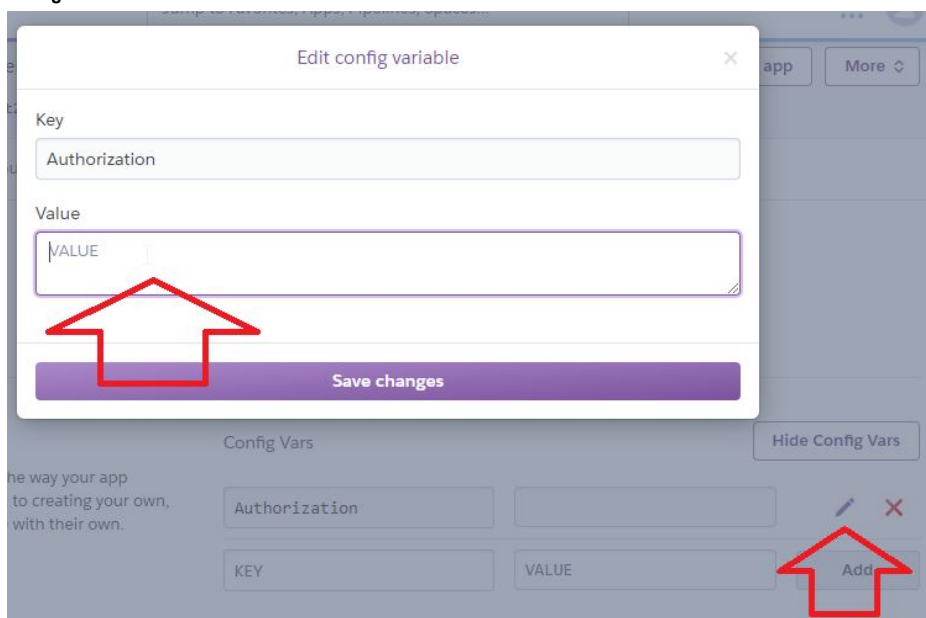
7. ไปที่เว็บ heroku.com ล็อกอิน แล้วคลิกที่ชื่อ app



8. กดที่ Setting แล้วกดที่ Reveal Config Vars

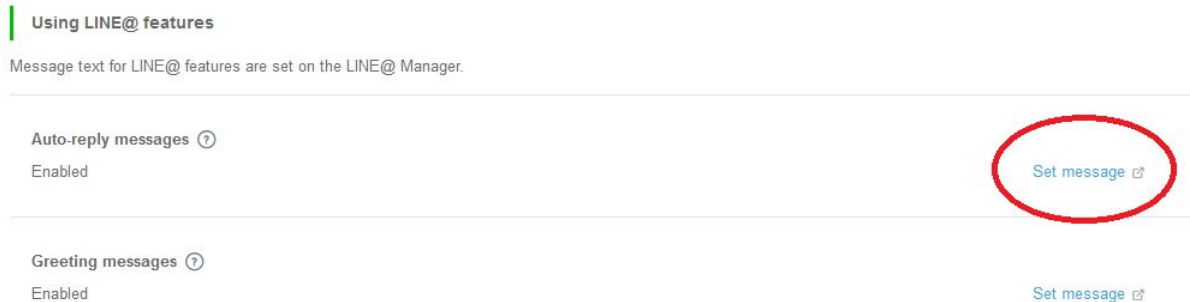


9. กดรูปดินสอ (Edit) วางค่าที่คัดลอกมา ในช่อง Value แล้วกด Save changes



การปรับแต่ง และทดสอบ LineBot

1. กลับไปที่เว็บ Line Developers ในส่วน Channel setting เลื่อนลงด้านล่าง หัวข้อ Auto-reply messages ให้กดที่ Set message (ด้านขวามือ)

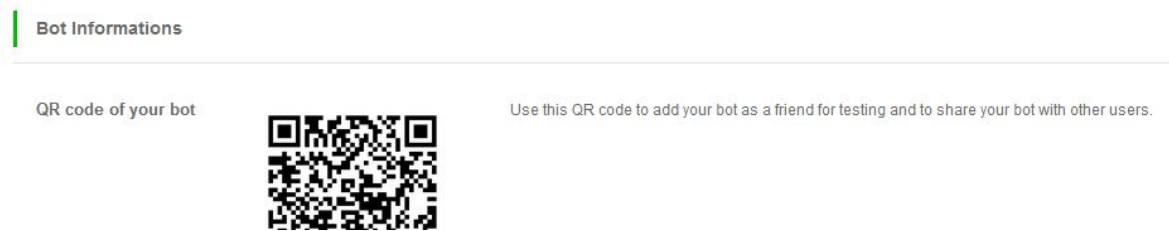


2. จะเปิดหน้าต่างเว็บ manager.line.biz ขึ้นมา ให้ล็อกอินด้วยบัญชีไลน์เช่นเดิม แล้วตั้งค่าดังภาพ (ปิด - ตอบกลับ, เปิด - Webhook)

ตั้งค่าการตอบกลับ

โหมดการตอบกลับ ?	<input checked="" type="radio"/> บอດ
	<input type="radio"/> แชนท
ข้อความทักทายเพื่อนใหม่	<input checked="" type="radio"/> เปิด
	<input type="radio"/> ปิด
	ตั้งค่าข้อความทักทายเพื่อนใหม่
ข้อความตอบกลับอัตโนมัติ	<input type="radio"/> เปิด
	<input checked="" type="radio"/> ปิด
	ตั้งค่าข้อความตอบกลับอัตโนมัติ
Webhook	<input checked="" type="radio"/> เปิด
	<input type="radio"/> ปิด
	ตั้งค่า Messaging API

3. กลับไปหน้าต่าง Line Developers อีกครั้ง เลื่อนลงมาหา QR code of your bot จากนั้นใช้มือถือสแกนเพื่อเพิ่มเพื่อน Line

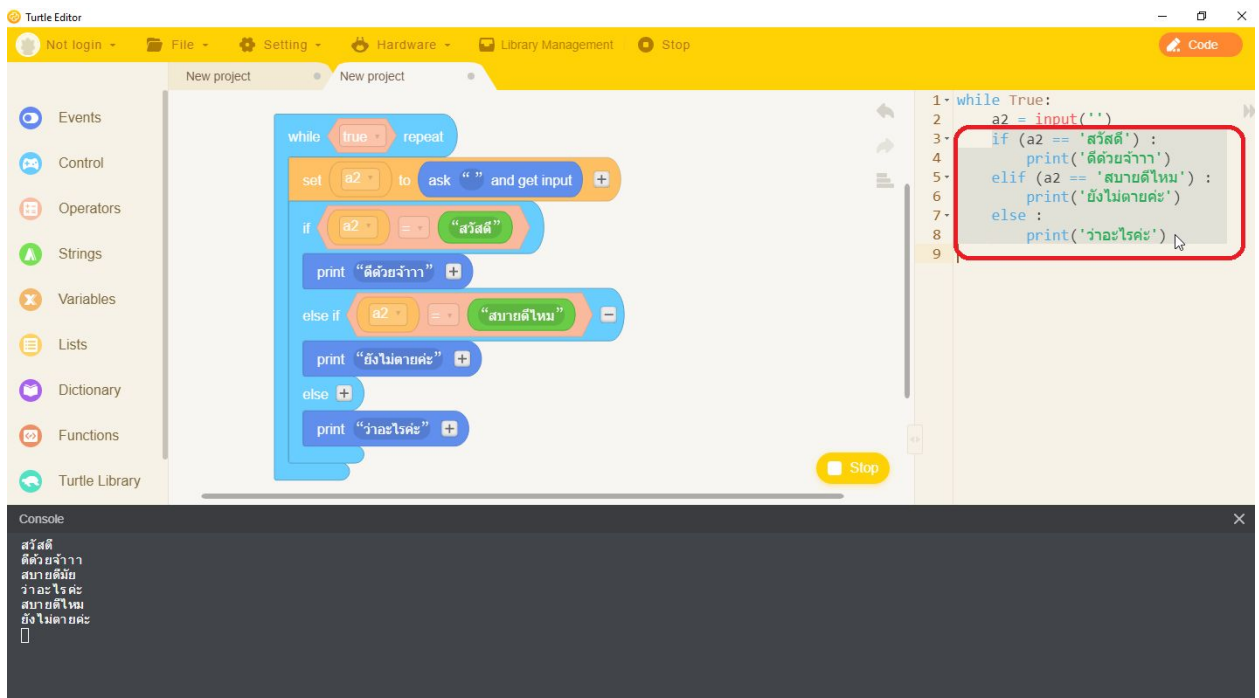


4. ทดลองส่งข้อความถึง LineBot ของเรา ซึ่งโค้ดเบื้องต้นของ bot จะตอบกลับด้วยข้อความที่เราส่งไป (ดังภาพ)



ปรับปรุงให้ bot สามารถโต้ตอบได้

1. เขียนคำสั่งเพื่อตรวจสอบข้อความที่ส่งมาหา bot เพื่อโต้ตอบ ในที่นี้เพื่อให้ง่ายต่อการโค้ดดิ้ง เราจะใช้ Turtle editor ซึ่งเป็นเครื่องมือเขียนโปรแกรมภาษาไพธอนแบบบล็อก เนื่องจากใช้วิธีลากวาง จึงไม่ต้องจดจำคำสั่งและรูปแบบการใช้งานคำสั่ง แต่จำเป็นต้องแก้ไขคำสั่งตอนนำไปใช้งานบ้าง จากนั้นคัดลอกคำสั่ง (เฉพาะส่วนที่เกี่ยวข้องกับการตรวจสอบข้อความ-ดังรูป) ภาษาไพธอน



2. ที่ GitHub ให้แก้ไขไฟล์ app.py โดยมองหาคำสั่ง `sendText(user, userText)`

```
17
18 @app.route('/callback', methods=['POST'])
19 def callback():
20     json_line = request.get_json()
21     json_line = json.dumps(json_line)
22     decoded = json.loads(json_line)
23     user = decoded["events"][0]['replyToken']
24     userText = decoded["events"][0]['message']['text']
25     sendText(user, userText)
26     return '', 200
27
```

3. วางคำสั่งที่คัดลอกจาก Turtle editor แทนที่คำสั่ง sendText(user, userText)

```
18 @app.route('/callback', methods=['POST'])
19 def callback():
20     json_line = request.get_json()
21     json_line = json.dumps(json_line)
22     decoded = json.loads(json_line)
23     user = decoded["events"][0]['replyToken']
24     userText = decoded["events"][0]['message']['text']
25     if (a2 == 'สวัสดี') :
26         print('ดีด้วยจ้าา')
27     elif (a2 == 'สบายดีไหม') :
28         print('ยังไม่ตายค่ะ')
29     else :
30         print('ว่าอะไรค่ะ')
31     return '',200
```

4. แก้ไขชื่อตัวแปรจาก Turtle editor (ในที่นี้คือ a2) เป็น userText และคำสั่ง print(แก้เป็น sendText(user,

```
17
18 @app.route('/callback', methods=['POST'])
19 def callback():
20     json_line = request.get_json()
21     json_line = json.dumps(json_line)
22     decoded = json.loads(json_line)
23     user = decoded["events"][0]['replyToken']
24     userText = decoded["events"][0]['message']['text']
25     if (userText == 'สวัสดี') :
26         sendText(user, 'ดีด้วยจ้าา')
27     elif (userText == 'สบายดีไหม') :
28         sendText(user, 'ยังไม่ตายค่ะ')
29     else :
30         sendText(user, 'ว่าอะไรค่ะ')
31     return '',200
32
```

5. ที่แอปไลน์ให้ทดสอบโดยการส่งข้อความไปหา bot (น้องฟ้า) ซึ่งต้องโต้ตอบตามเงื่อนไขที่ตั้งไว้

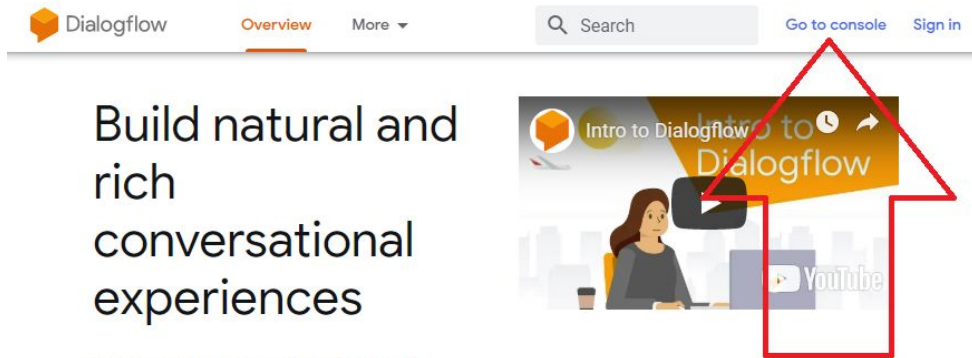
ปรับปรุงการตรวจสอบข้อความให้แม่นยำ ด้วย DialogFlow

การรับข้อความจากไลน์โดยตรงมาตรวจสอบ เราต้องเขียนเงื่อนไขในการตรวจสอบให้ครอบคลุมคำที่อาจใช้ในเจตนาที่สื่อความในความหมายเดียวกัน ซึ่งจะทำให้โค้ดของเรายาวมาก แถมอาจไม่ครบถ้วน ดังตัวอย่าง

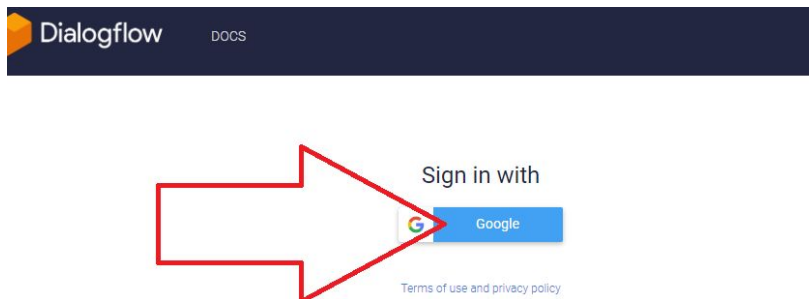


การใช้ AI มาช่วยวิเคราะห์ข้อความ จึงเป็นทางเลือกที่ดี ในที่นี้เราเลือกบริการของ Google ที่ชื่อว่า DialogFlow

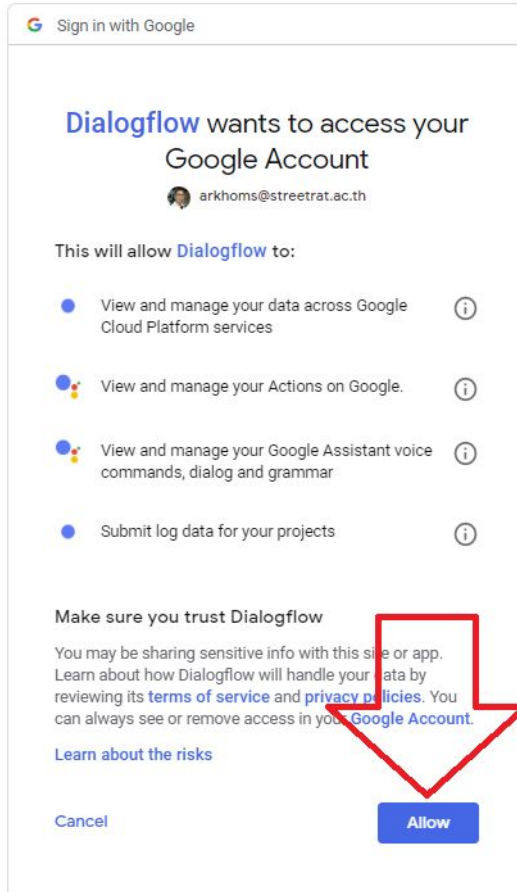
1. ไปที่เว็บไซต์ dialogflow.com แล้วคลิกที่ Go to console



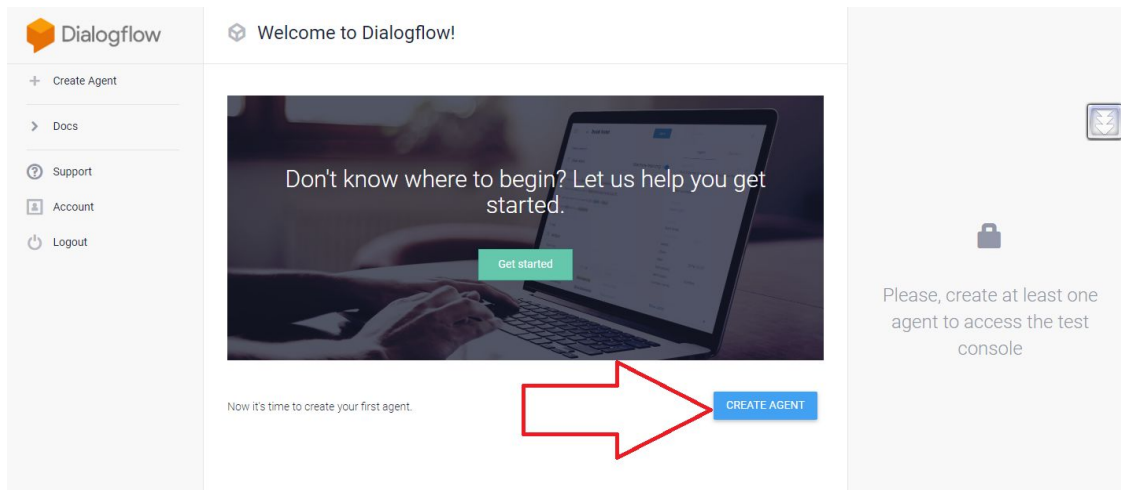
2. กดที่ ปุ่ม Google



3. กด Allow เพื่ออนุญาตให้ DialogFlow ใช้ข้อมูลในบัญชี Google



4. กด CREATE AGENT เพื่อสร้าง Agent (เปรียบเสมือนตัวแทนของ bot)



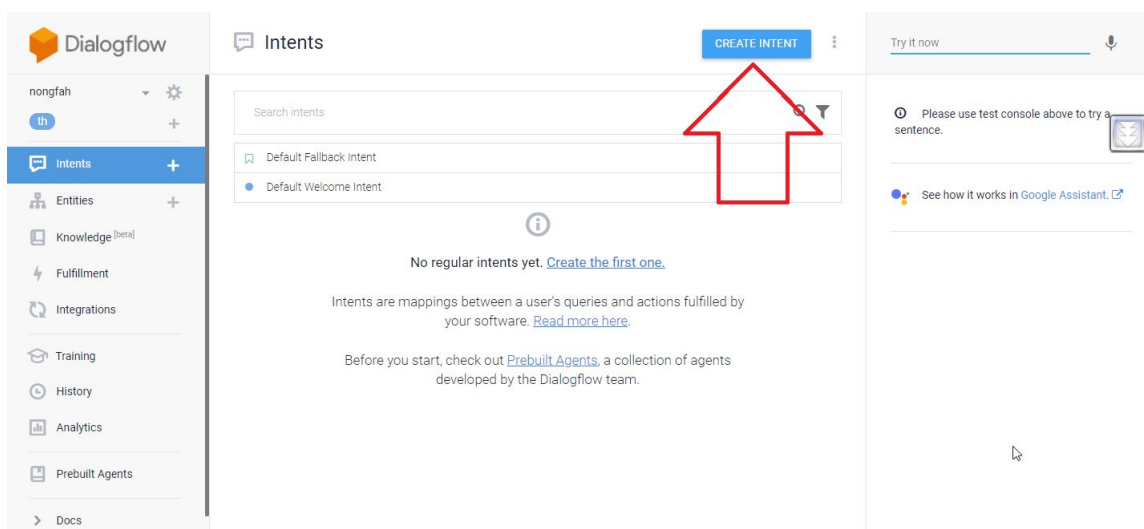
5. ตั้งชื่อ Agent และเลือกภาษาตั้งต้น

nongfah CREATE

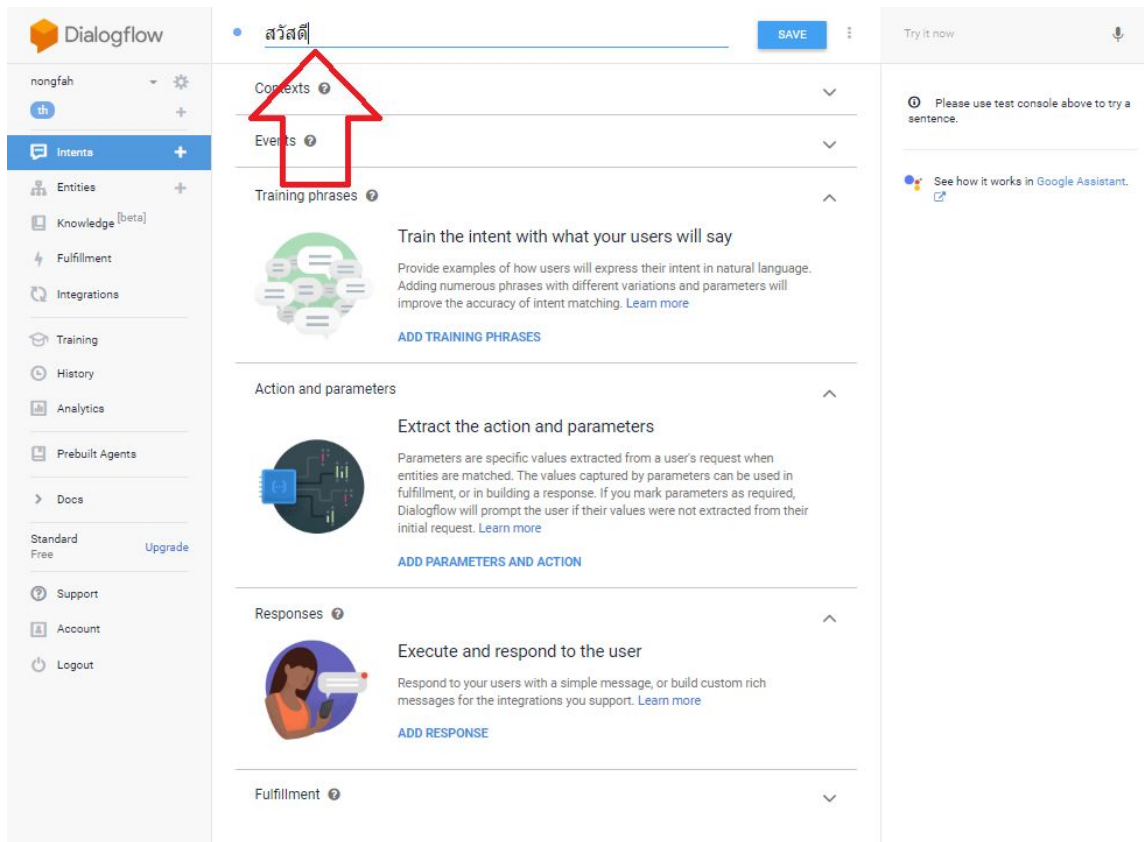
DEFAULT LANGUAGE th Thai - th (GMT+7:00) Asia/Bangkok

Google Project Create a new Google project

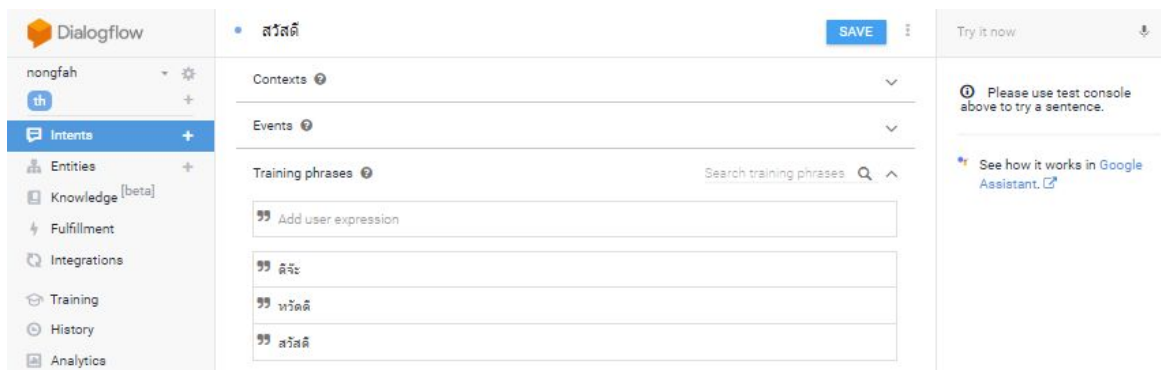
6. กด CREATE INTENT เพื่อกำหนดข้อความที่ต้องการสื่อความ



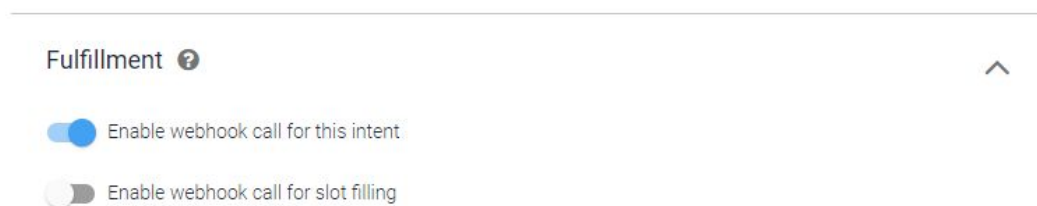
7. กำหนดข้อความที่ต้องการสื่อความ



8. กดที่ ADD TRAINING PHRASES เพื่อกำหนดข้อความที่เป็นไปได้ กรณีที่จะสื่อความตามข้อความในข้อ 7

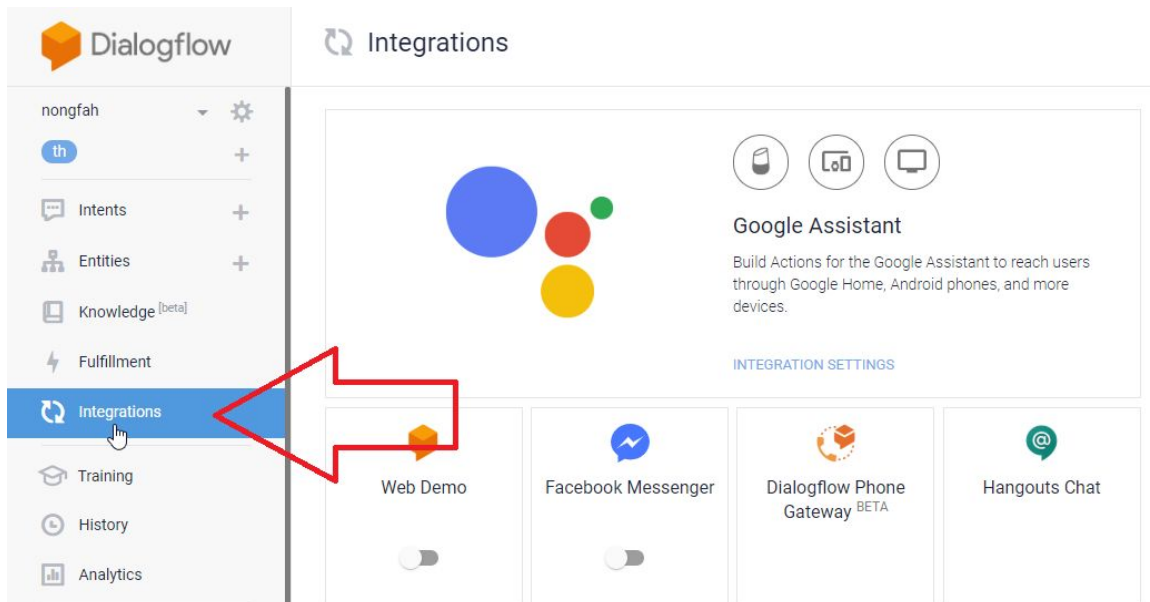


9. เลื่อนลงไปที่หัวข้อ Fulfillment กรณีต้องการส่งข้อความไปยัง bot ของเรา

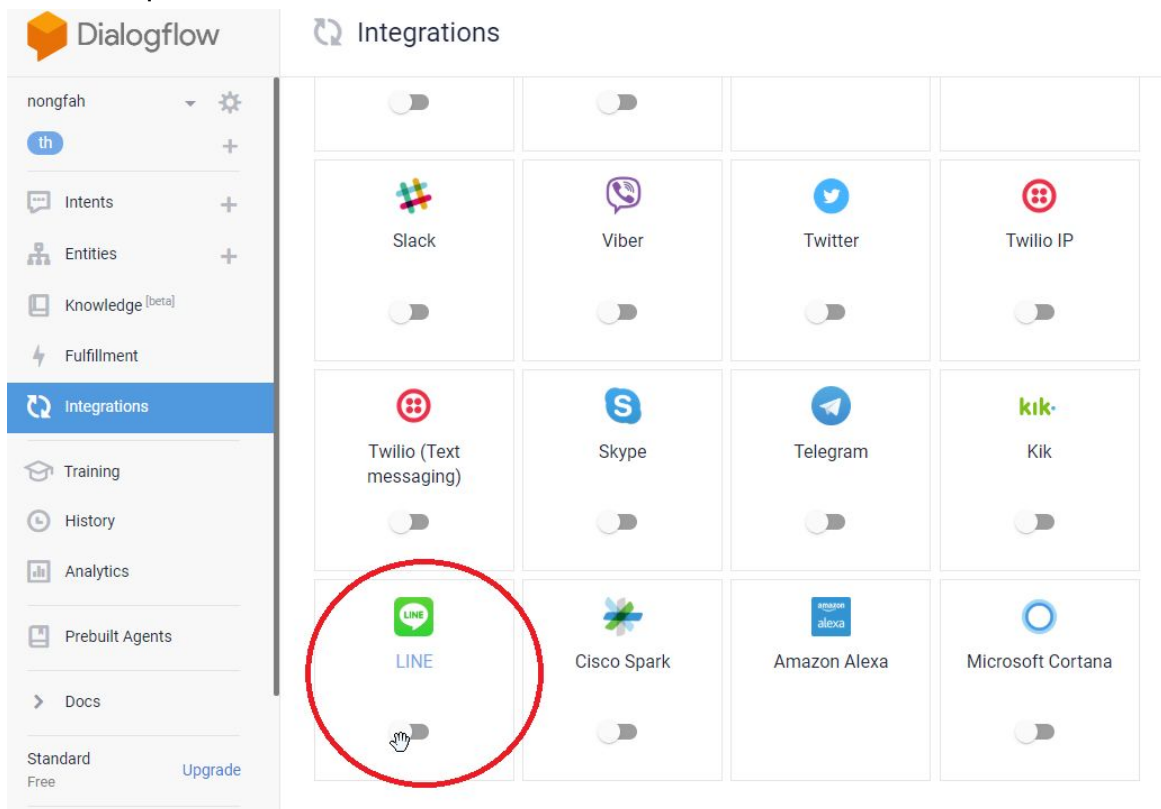


10. จากนั้น กด SAVE (ด้านบน)

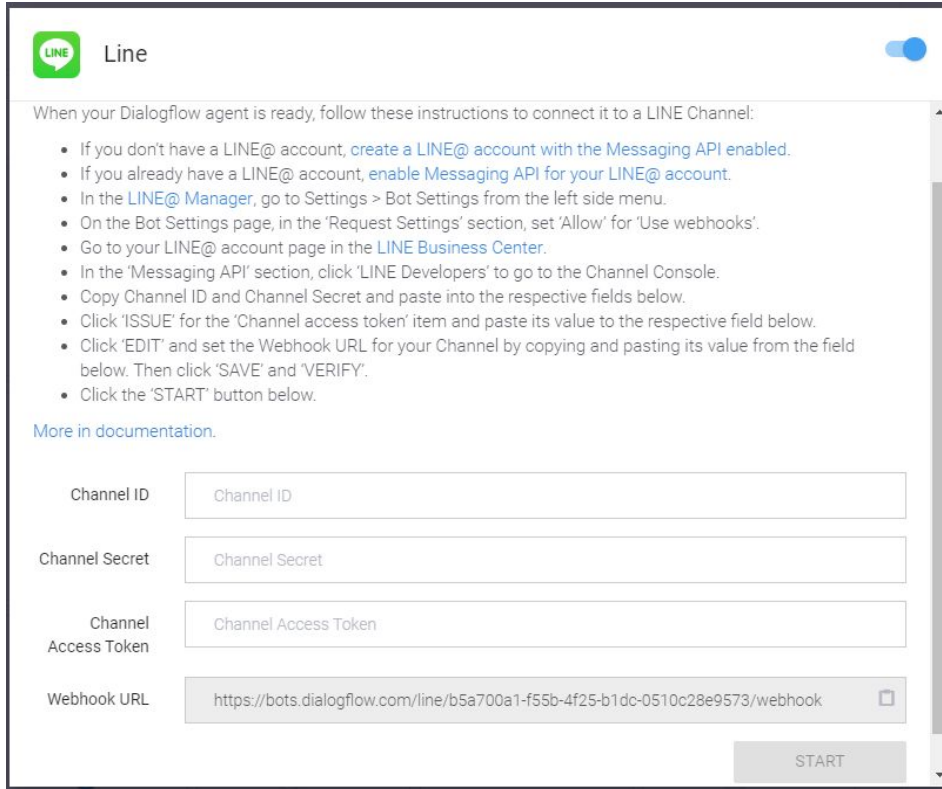
11. ที่เมนูด้านซ้าย (หากทำบนมือถือ อาจต้องกดที่ ชิด 3 ชิด) เลือก Integrations



12. เลื่อนปุ่มสไลด์ใต้ไอคอน LINE (เพื่อกำหนดค่าให้เชื่อมกับบัญชี LineAPI)



13. ที่หน้าตาแบบนี้ ให้กรอกข้อมูล (จากเว็บ developers.line.me) ใส่ให้ครบถ้วน (Channel ID, Channel Secret และ Channel Access Token) แล้วกดปุ่ม START



Line

When your Dialogflow agent is ready, follow these instructions to connect it to a LINE Channel:

- If you don't have a LINE@ account, [create a LINE@ account with the Messaging API enabled](#).
- If you already have a LINE@ account, [enable Messaging API for your LINE@ account](#).
- In the [LINE@ Manager](#), go to Settings > Bot Settings from the left side menu.
- On the Bot Settings page, in the 'Request Settings' section, set 'Allow' for 'Use webhooks'.
- Go to your LINE@ account page in the [LINE Business Center](#).
- In the 'Messaging API' section, click 'LINE Developers' to go to the Channel Console.
- Copy Channel ID and Channel Secret and paste into the respective fields below.
- Click 'ISSUE' for the 'Channel access token' item and paste its value to the respective field below.
- Click 'EDIT' and set the Webhook URL for your Channel by copying and pasting its value from the field below. Then click 'SAVE' and 'VERIFY'.
- Click the 'START' button below.

[More in documentation.](#)

Channel ID

Channel Secret

Channel Access Token

Webhook URL

START

14. คัดลอก Webhook URL จาก DialogFlow (กดไอคอนสี่เหลี่ยมด้านขวา) เพื่อนำไปแก้ไขในเว็บ developers.line.me หัวข้อ Webhook URL (ตอนวางแล้วอย่าลืมลบ https:// ด้านหน้าออก เพื่อจะได้ไม่ซ้ำกับที่ line เตรียมไว้ให้)

Webhook URL Requires SSL ?

https://

Max: 500 characters

Please enter a URL that supports HTTPS.

Update

Cancel

15. กด Verify เพื่อตรวจสอบการเชื่อมต่อ

Webhook URL Requires SSL ?

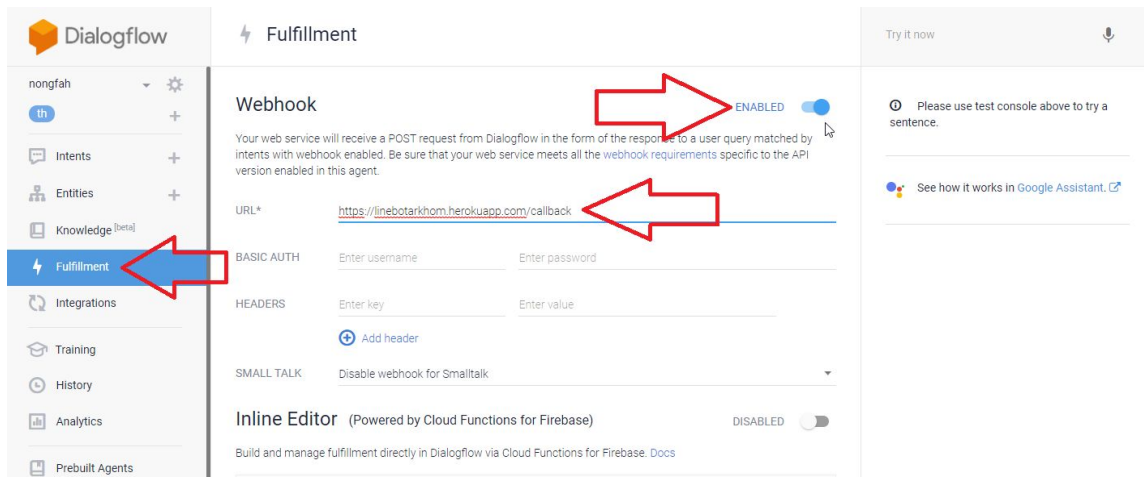
https://bots.dialogflow.com/line/b5a700a1-f55b-4f25-b1dc-0510c28e9573/webhook

Verify

Edit

✓ Success

16. ไปที่เว็บ dialogFlow คลิกที่เมนู Fulfillment (ด้านซ้าย) กำหนดให้เป็น ENABLED แล้วป้อน Webhook URL ที่เคยกรอกในไลน์ ในช่อง URL แล้วกด SAVE (ด้านล่าง)



17. ที่เว็บ GitHub ให้แก้ไขคำสั่งบรรทัดที่ 23-24 โดยแก้จาก `user = decoded["events"][0]["replyToken"]` เป็น `user = decoded['originalDetectIntentRequest']['payload']['data']['replyToken']` และ `userText = decoded["events"][0]["message"]["text"]` เป็น `userText = decoded['queryResult']['intent']['displayName']`

```
18 @app.route('/callback', methods=['POST'])
19 def callback():
20     json_line = request.get_json()
21     json_line = json.dumps(json_line)
22     decoded = json.loads(json_line)
23     user = decoded['originalDetectIntentRequest']['payload']['data']['replyToken'] #user = decoded["events"][0]["replyToken"]
24     userText = decoded['queryResult']['intent']['displayName'] #userText = decoded["events"][0]["message"]["text"]
25     if (userText == 'สวัสดี') :
26         sendText(user, 'สวัสดีจ้าว')
27     elif (userText == 'สวัสดีใหม่') :
28         sendText(user, 'ยังไม่เคยคุย')
29     else :
30         sendText(user, 'ทำอะไรคะ')
31     return '',200
32
```

18. ทดลองส่งข้อความถึง “น้องฟ้า” อีกครั้ง

