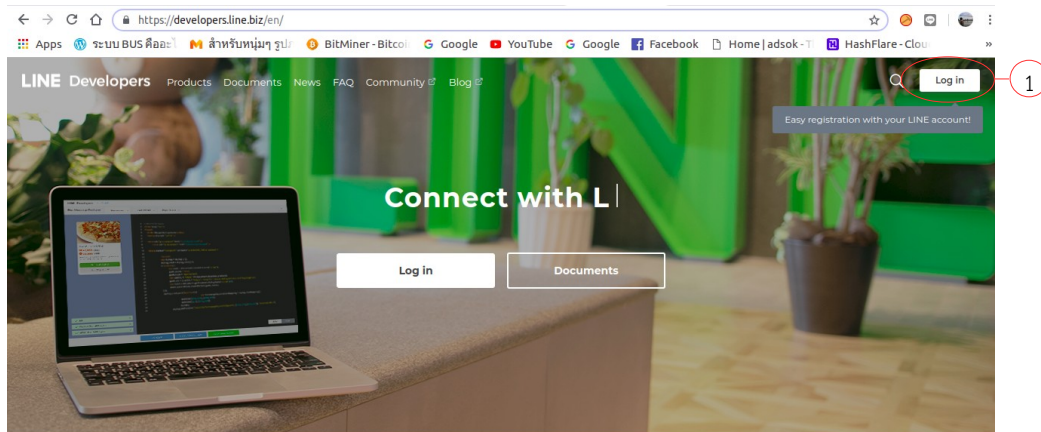


Line Beacon

1 การสร้าง LINE Bot

1.1 การสร้าง LINE Bot

การที่จะได้ LINE Bot จำเป็นจะต้องมี ID LINE จากนั้นไปที่เว็บ <https://developers.line.biz/en/> ทำการ Login เพื่อสร้าง LINE Bot

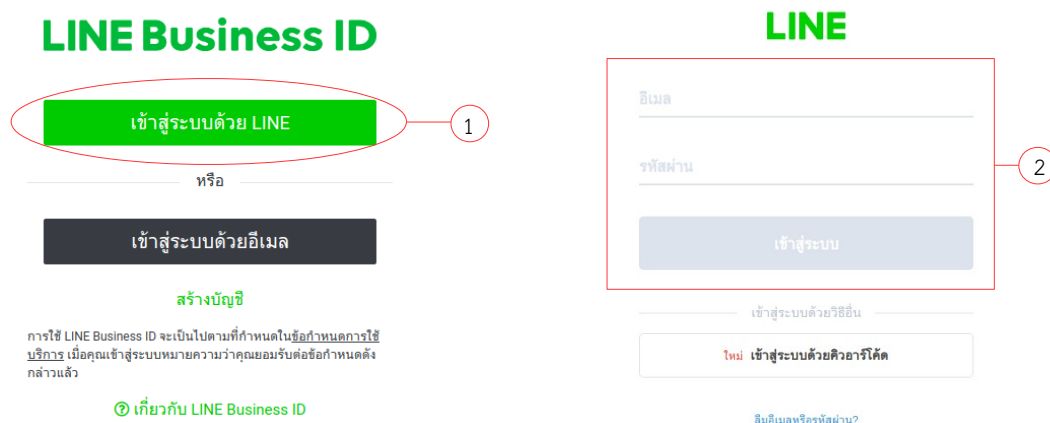


Products

รูปที่ 1.1 แสดงแผนภาพ <https://developers.line.biz/en/>

1.2 เข้าสู่ระบบด้วย ID LINE

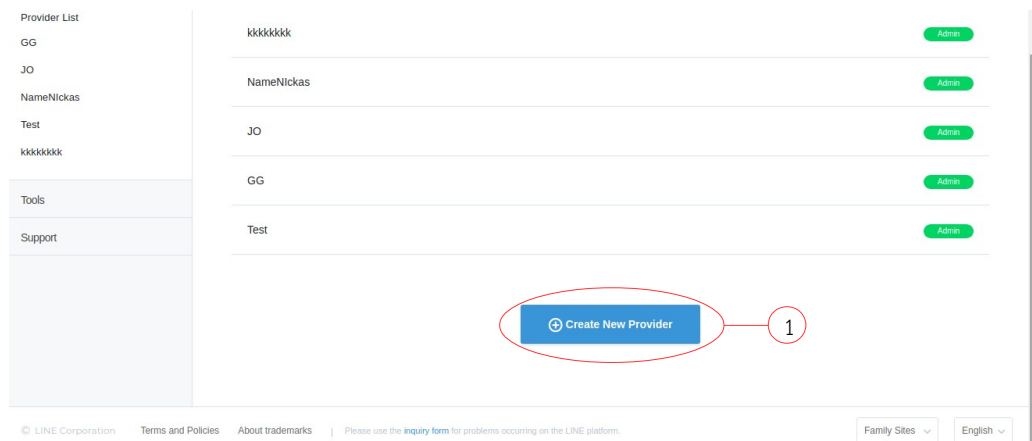
ทำการ Login โดยใช้ ID LINE เพื่อสร้าง LINE Bot



รูปที่ 1.2 แสดงแผนภาพการเข้าสู่ระบบด้วย ID LINE

1.3 ทำการสร้าง LINE Bot

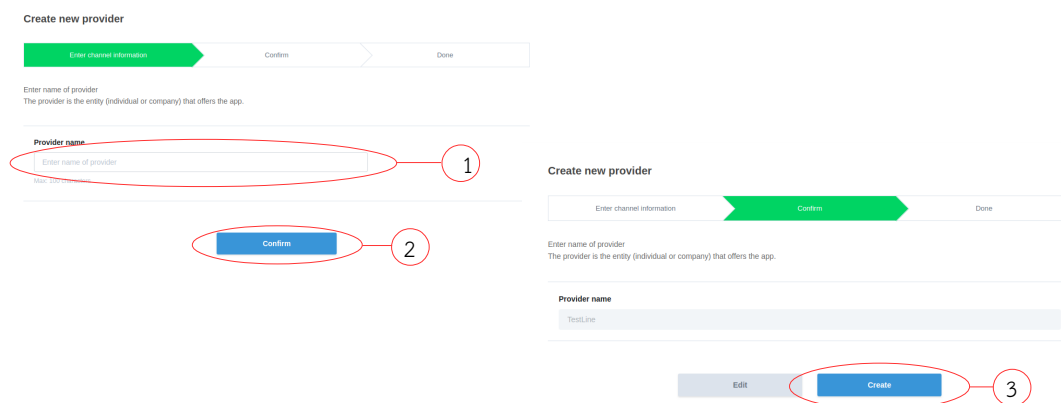
กดที่ Create New Provider สร้างผู้ให้บริการ



รูปที่ 1.3 แสดงแผนภาพ สร้าง LINE Bot

1.4 ทำการสร้าง LINE Bot

ตั้งชื่อผู้ให้บริการ จากนั้นกดที่ Confirm จะให้ดูอีกครั้งเพื่อยืนยันความถูกต้อง จากนั้นกด Create



รูปที่ 1.4 แสดงแผนภาพ สร้าง LINE Bot

1.5 ทำการสร้าง LINE Bot

เมื่อสร้างผู้ให้บริการเสร็จสิ้น จะได้บริการ LINE Login, Messaging API และ Clova Skill ซึ่งที่ใช้ทำ LINE Bot คือ Messaging API จากนั้นเลือกประเภท และ กรอก Email กด Confirm

TestLine

A provider has been created!
Next, let's create a new channel

Please select a channel from the following and create a new channel.

LINE Login
Create Channel

Messaging API
Create Channel

Clova Skill
Create Channel

Plan
Developer Trial
A trial plan which lets you create a bot that can send push messages and have up to 50 friends.
Note: You cannot upgrade or buy a premium ID for a Developer Trial plan.

Category
เว็บไซต์บล็อก

Subcategory
เว็บไซต์บล็อก(อื่นๆ)

Email address
example@line.me
Max: 100 characters

Previous page

Confirm

รูปที่ 1.5 แสดงแผนภาพ สร้าง LINE Bot

1.6 ทำการสร้าง LINE Bot

เมื่อกด Confirm แล้วให้ติ๊กข้อตกลงทั้ง 2 จากนั้นกดไปที่ Create

Category
เว็บไซต์บล็อก

Subcategory
เว็บไซต์บล็อก(อื่นๆ)

Email address
nictheripper@gmail.com

1

☒ LINE@ Terms of Use I have read and agree to the Terms of Use.

☒ Messaging API (Developer Trial plan) Terms of Use: I have read and agree to the Terms of Use.

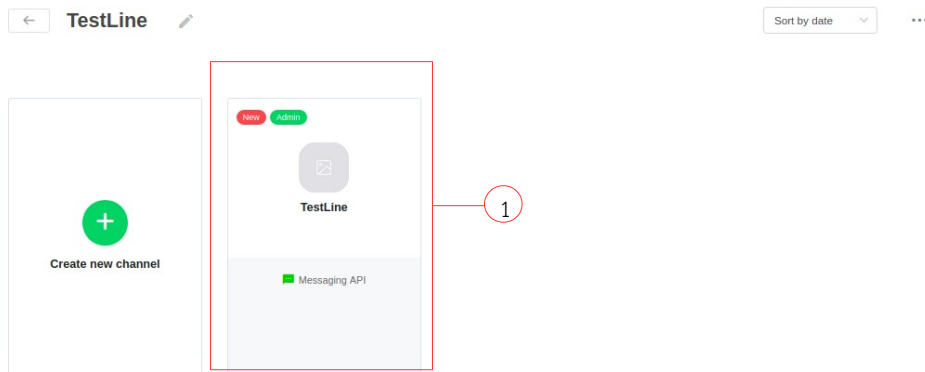
Edit

Create

2

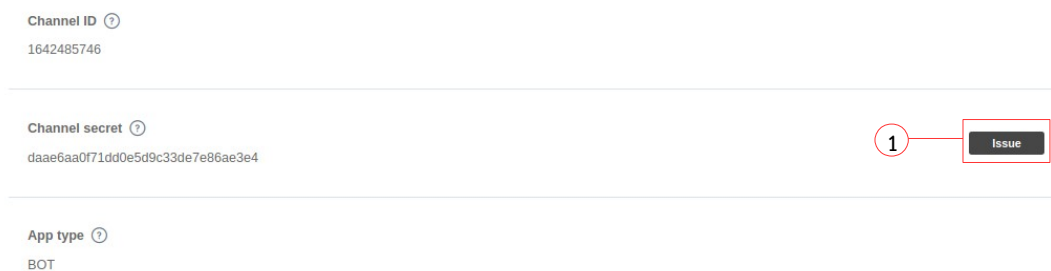
รูปที่ 1.6 แสดงแผนภาพ สร้าง LINE Bot

1.7 เสร็จสิ้นการสร้าง LINE Bot จากนั้นจะได้ Messaging API

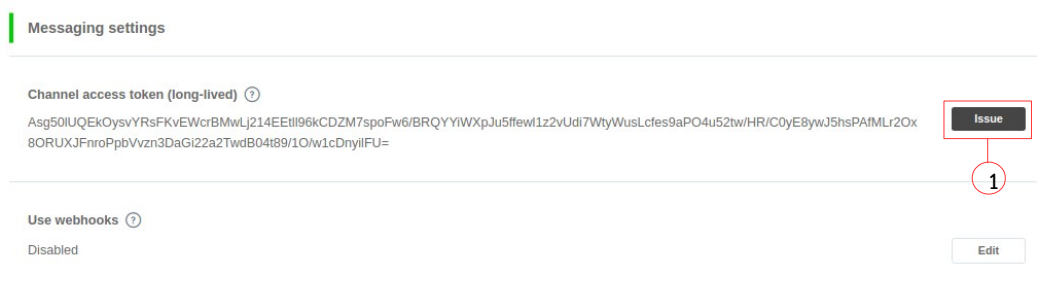


รูปที่ 1.7 แสดงแผนภาพ สร้าง LINE Bot

1.8 การขอ Channel secret และ Channel access token



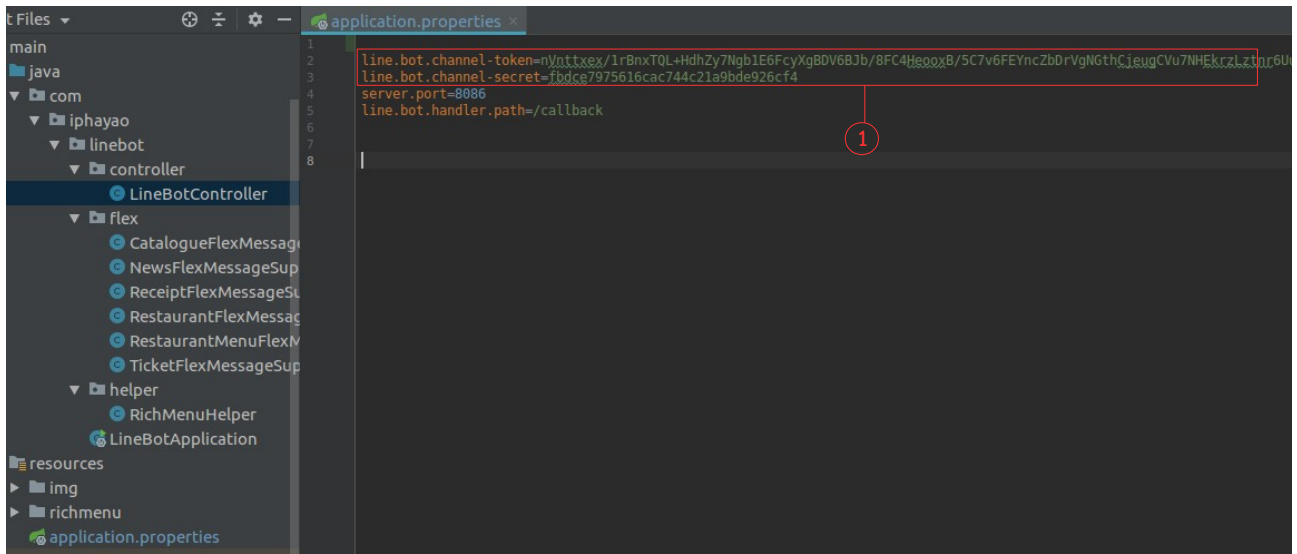
รูปที่ 1.8 แสดงแผนภาพ ขอ Channel secret



รูปที่ 1.9 แสดงแผนภาพ ขอ Channel access token

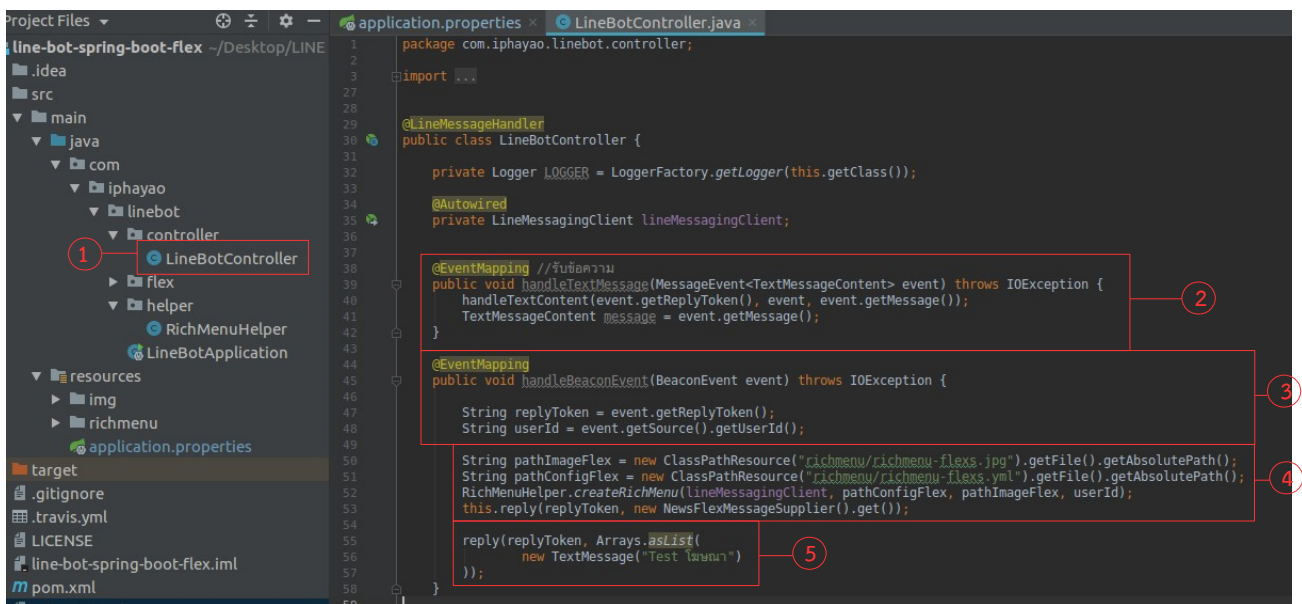
2. โหลดโปรแกรม git clone <https://github.com/epsonthenic/line-spring-beacon.git>

2.1 เมื่อได้ Channel secret และ Channel access token นำมาใส่ในโปรแกรม



รูปที่ 2.1 แสดงแผนภาพ การใส่ Channel secret และ Channel access token

2.2 ไปที่ controller class LineBotController



รูปที่ 2.2 แสดงแผนภาพ LineBotController

2.2.1 class LineBotController

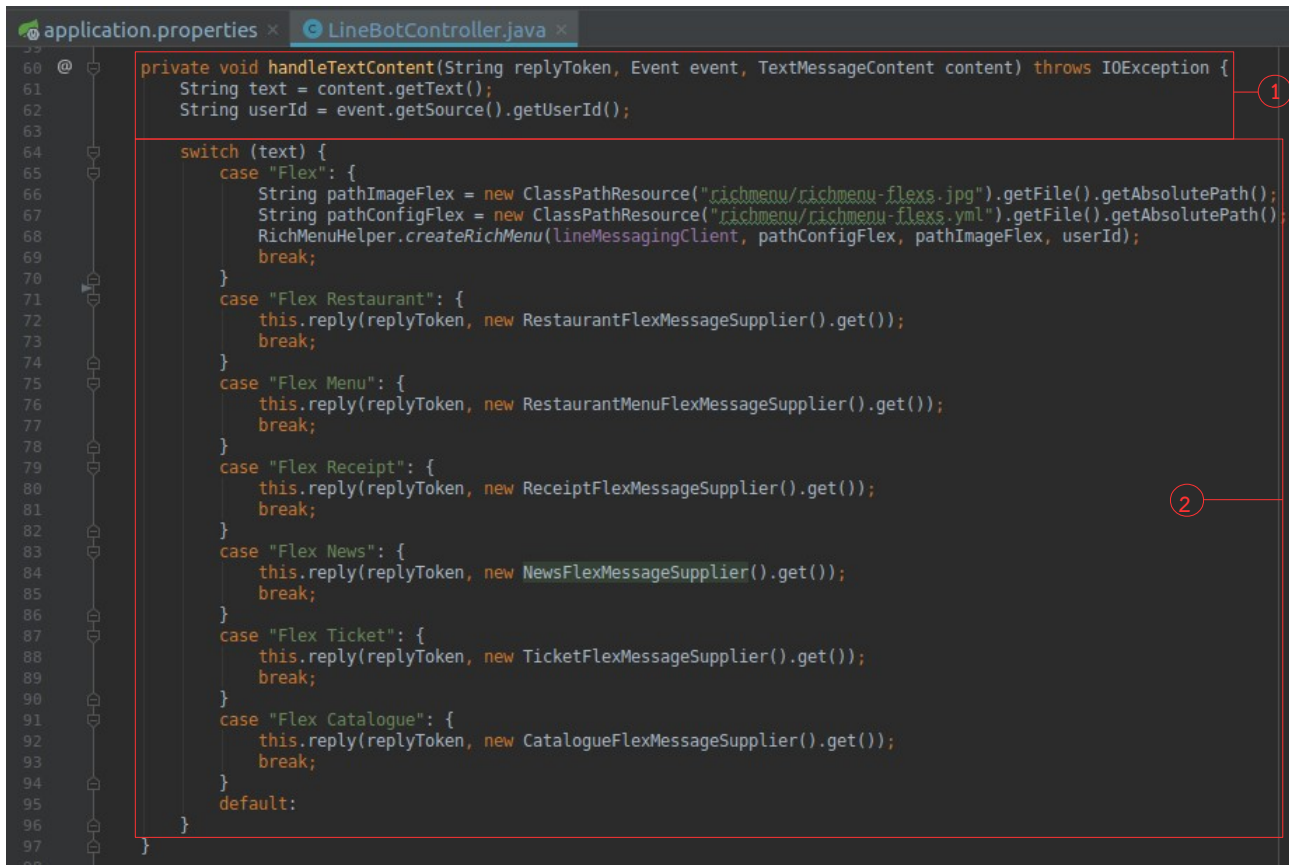
2.2.2 เป็นเหตุการณ์เมื่อผู้ใช้ส่งข้อความหา LINE Bot จากนั้น LINE Bot จะทำการ get ข้อความ, Token

2.2.3 เป็นเหตุการณ์เมื่อผู้ใช้อยู่ในระยะของ Bluetooth จากนั้น LINE Bot จะทำการ get LINE id, Token

2.2.4 นำ รูปของ Menu มาเก็บไว้ที่ pathImageFlex และ นำ Menu มาเก็บไว้ที่ pathConfigFlex จากนั้นเอา Menu ไปแสดงใน LINE Bot และ แสดงหน้าข่าวเป็น Rich Menu

2.2.5 แสดงข้อความใน LINE Bot ว่า Test โฆษณา

2.3 LINE Bot ตอบกลับ



```
application.properties x LineBotController.java x
60 @
61 private void handleTextContent(String replyToken, Event event, TextMessageContent content) throws IOException {
62     String text = content.getText();
63     String userId = event.getSource().getUserId();
64
65     switch (text) {
66         case "Flex": {
67             String pathImageFlex = new ClassPathResource("richmenu/richmenu-flexs.jpg").getFile().getAbsolutePath();
68             String pathConfigFlex = new ClassPathResource("richmenu/richmenu-flexs.yml").getFile().getAbsolutePath();
69             RichMenuHelper.createRichMenu(lineMessagingClient, pathConfigFlex, pathImageFlex, userId);
70             break;
71         }
72         case "Flex Restaurant": {
73             this.reply(replyToken, new RestaurantFlexMessageSupplier().get());
74             break;
75         }
76         case "Flex Menu": {
77             this.reply(replyToken, new RestaurantMenuFlexMessageSupplier().get());
78             break;
79         }
80         case "Flex Receipt": {
81             this.reply(replyToken, new ReceiptFlexMessageSupplier().get());
82             break;
83         }
84         case "Flex News": {
85             this.reply(replyToken, new NewsFlexMessageSupplier().get());
86             break;
87         }
88         case "Flex Ticket": {
89             this.reply(replyToken, new TicketFlexMessageSupplier().get());
90             break;
91         }
92         case "Flex Catalogue": {
93             this.reply(replyToken, new CatalogueFlexMessageSupplier().get());
94             break;
95         }
96         default:
97     }
98 }
```

The screenshot shows the `LineBotController.java` file in an IDE. The `handleTextContent` method is highlighted with a red box labeled '1'. Inside the method, a switch statement handles different text inputs. The 'Flex' case is highlighted with a red box labeled '2', showing the logic to create a rich menu using `RichMenuHelper.createRichMenu`. Other cases like 'Flex Restaurant', 'Flex Menu', 'Flex Receipt', 'Flex News', 'Flex Ticket', and 'Flex Catalogue' are also shown, each calling a specific `MessageSupplier` to generate a response.

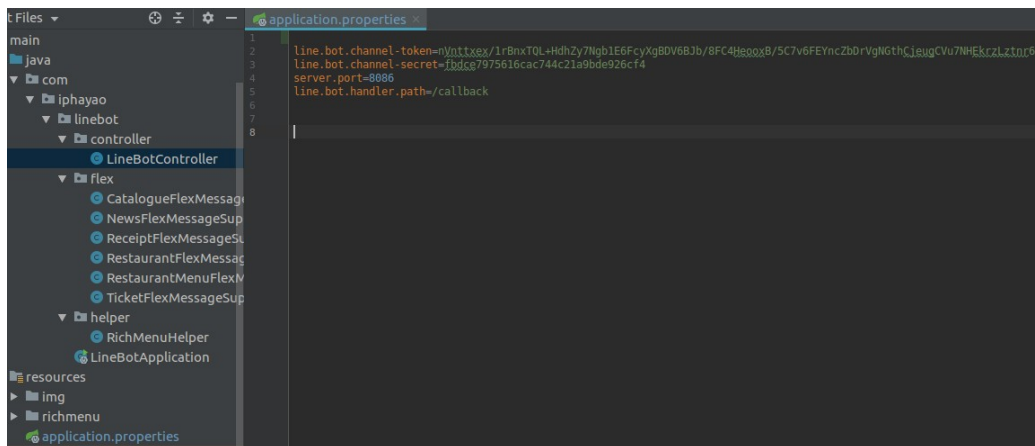
รูปที่ 2.2 แสดงแผนภาพ LineBotController

2.2.1 เมื่อผู้ใช้ส่งข้อความหา LINE Bot จะทำการ get ข้อความ, LINE id

2.2.2 เช็คว่าข้อความที่ได้ตรงเงื่อนไขไหนให้ทำเงื่อนไขนั้น ถ้าไม่ตรงตามเงื่อนไข จบการทำงานของเหตุการณ์รับข้อความ

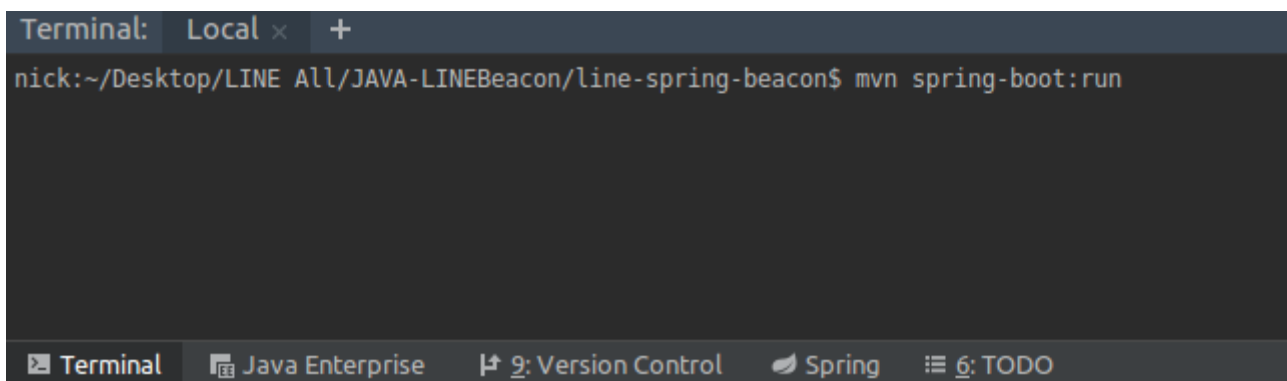
3. Run LINE Bot

3.1 ตั้งค่าต่างๆได้ที่ Application.properties

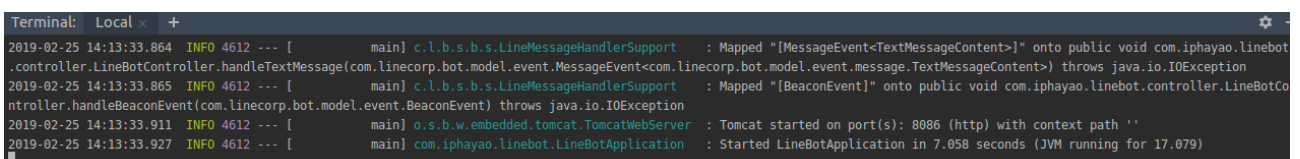


รูปที่ 3.1 แสดงแผนภาพ LineBotController

3.2 ไปที่หน้า Terminal เพื่อ Run



รูปที่ 3.2 แสดงแผนภาพ mvn spring-boot:run



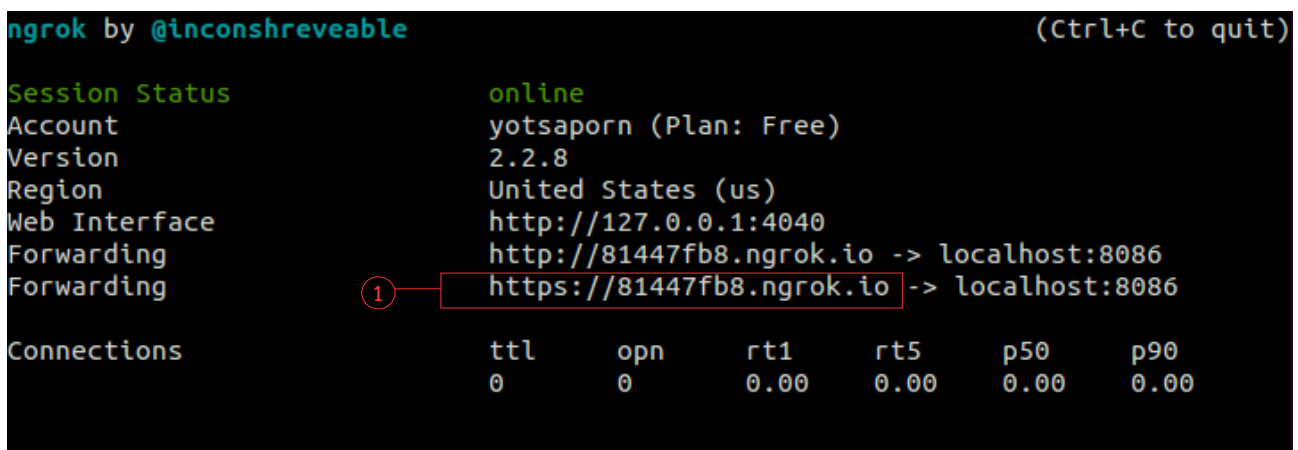
รูปที่ 3.3 แสดงแผนภาพ port ที่ได้ 8086

4. ทำให้ LINE Bot Online

ทำการติดตั้ง ngrok เพื่อให้ Online

```
$ mkdir ngrok
$ cd ngrok/
$ wget -c https://bin.equinox.io/c/4VmDzA7iaHb/ngrok-stable-linux-amd64.zip
$ unzip ngrok-stable-linux-amd64.zip
$ ./ngrok http 8086
```

รูปที่ 4.1 แสดงแผนภาพ ติดตั้ง ngrok



รูปที่ 4.2 แสดงแผนภาพ Run ./ngrok http 8086

4.1 ตั้งค่าให้กับ Messaging API

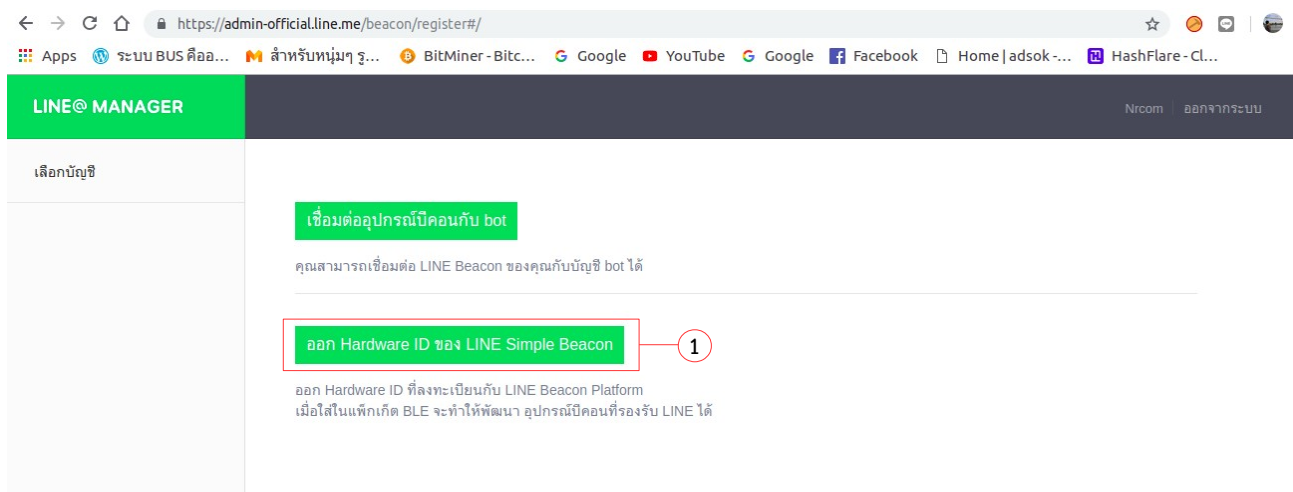
เพื่อให้ LINE Bot ทำงานกด Use webhooks ให้เป็น Enabled และ Webhook URL ให้นำ https ที่ได้จาก ngrok มาใส่ตามด้วย callback



รูปที่ 4.3 แสดงแผนภาพ ตั้งค่าให้กับ Messaging API

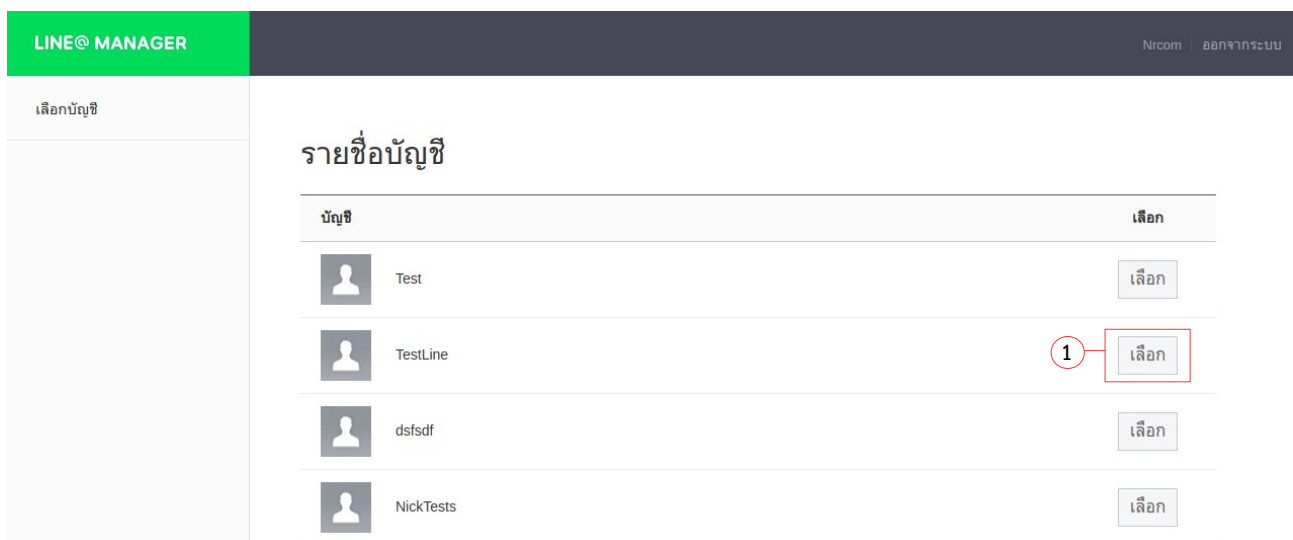
5. ขอ Hardware ID

5.1 ไปที่เว็บ <https://admin-official.line.me/beacon/register#/>



รูปที่ 5.1 แสดงแผนภาพ <https://admin-official.line.me/beacon/register#/>

5.2 เลือก LINE Bot ที่ให้เป็น Beacon



รูปที่ 5.2 แสดงแผนภาพ เลือก LINE Bot

5.3 ขอ Hardware ID

LINE@ MANAGER

เลือกบัญชี

ออก Hardware ID

Hardware ID ในแพ็คเกจ LINE Beacon สามารถออกใหม่ได้
แต่จะออก Hardware ID ได้ไม่เกิน 10 รายการ

ออก Hardware ID 1

ยังไม่ออก Hardware ID

ย้อนกลับ

รูปที่ 5.3 แสดงแผนภาพ ขอ Hardware ID

5.4 จะได้ Hardware ID

LINE@ MANAGER

เลือกบัญชี

ออก Hardware ID

Hardware ID ในแพ็คเกจ LINE Beacon สามารถออกใหม่ได้
แต่จะออก Hardware ID ได้ไม่เกิน 10 รายการ

ออก Hardware ID

1	012624594f 1	ลบ
---	--------------	----

ย้อนกลับ

รูปที่ 5.4 แสดงแผนภาพ ขอ Hardware ID

6. ติดตั้ง Nodejs

```
$ sudo apt-get install bluetooth bluez libbluetooth-dev libudev-dev  
$ curl -sL https://deb.nodesource.com/setup_6.x | sudo -E bash -  
$ sudo apt-get install nodejs
```

รูปที่ 6.1 แสดงแผนภาพ ติดตั้ง Nodejs v.6

7. โหลดโปรแกรม git clone <https://github.com/line/line-simple-beacon.git>

7.1 นำ Hardware ID ที่ได้มาใส่ --hwid= เพื่อเปิดใช้งาน LINE Beacon ผ่าน Bluetooth

```
$ cd line-simple-beacon/tools/line-simplebeacon-nodejs-sample/  
$ npm install  
  
หมายเหตุ ต้องเปิด Bluetooth ก่อนใส่คำสั่งถัดไป  
  
$ sudo ./simplebeacon.js --hwid=012624594f ①
```

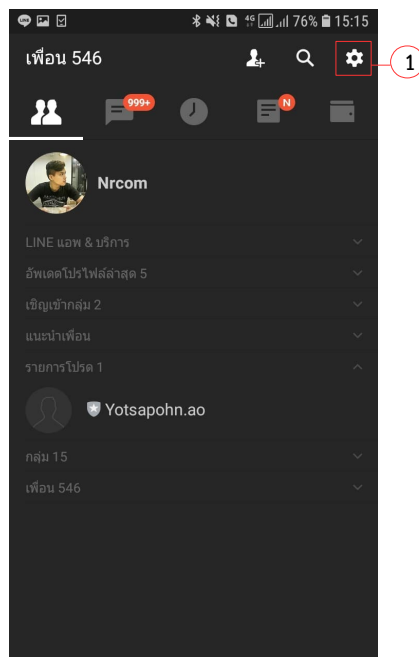
รูปที่ 7.1 แสดงแผนภาพ Run nodejs

```
nick:~/Desktop/LINE ALL/JAVA-LINEBeacon/line-simple-beacon/tools/line-simplebeacon-nodejs-sample$ sudo ./simplebeacon.js --hwid=012624594f  
stateChange: poweredOn  
<Buffer 02 01 06 03 03 6f fe 0b 16 6f fe 02 01 26 24 59 4f 7f 00>
```

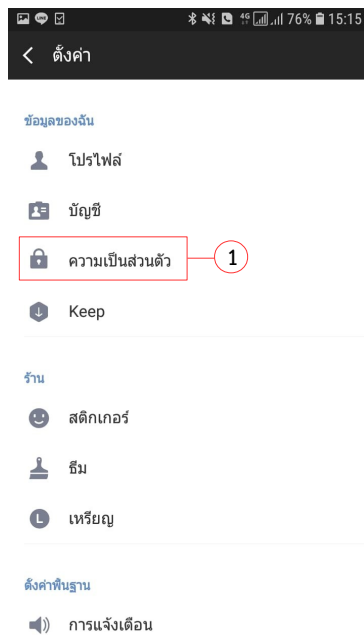
รูปที่ 7.2 แสดงแผนภาพ ถ้า Run ผ่าน จะหน้าต่างประมาณนี้

8. ทำการเพิ่ม LINE Bot เป็นเพื่อนใน LINE และเปิดใช้งาน LINE Beacon

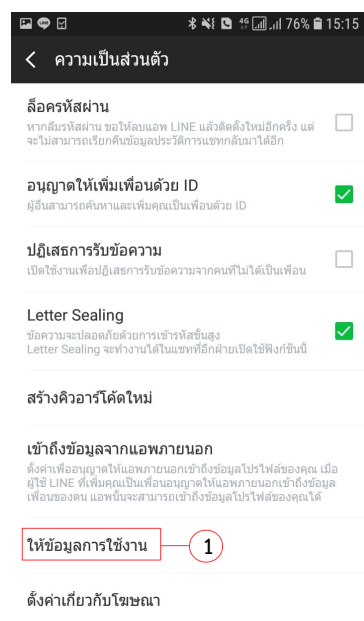
8.1 เพิ่ม LINE Bot เป็นเพื่อนใน LINE แล้วไปที่ตั้งค่า



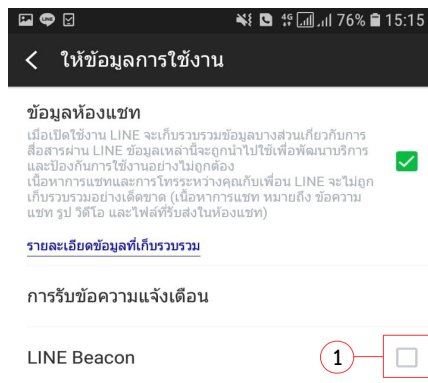
รูปที่ 8.1 แสดงแผนภาพ LINE



รูปที่ 8.2 แสดงแผนภาพ ตั้งค่า



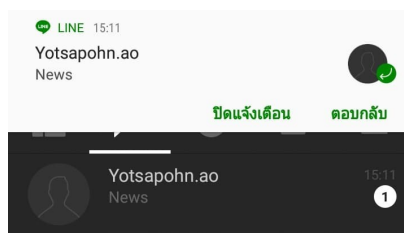
รูปที่ 8.3 แสดงแผนภาพ ตั้งค่าความเป็นส่วนตัว



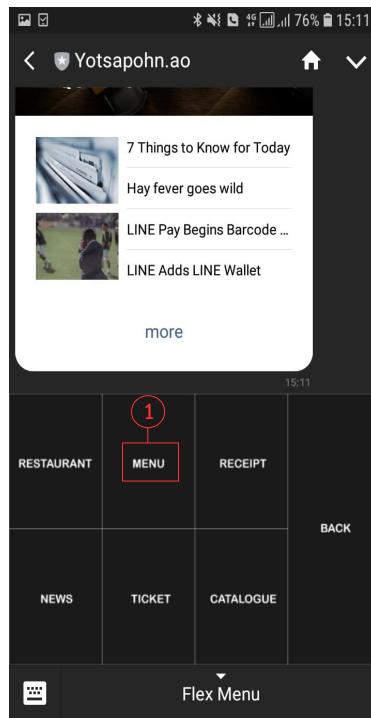
รูปที่ 8.4 แสดงแผนภาพ ให้ข้อมูลการใช้งาน



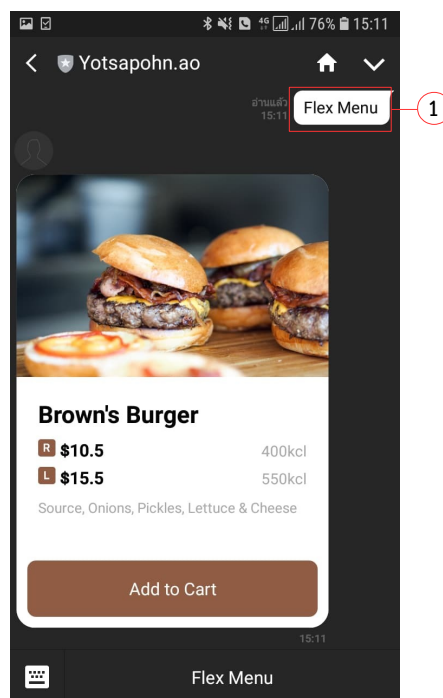
รูปที่ 8.5 แสดงแผนภาพ ยอมรับข้อกำหนด



รูปที่ 8.6 แสดงแผนภาพ การทำงานของ LINE Beacon อยู่ในระยะ



รูปที่ 8.7 แสดงแผนภาพ การทำงานของ LINE Beacon อยู่ในระยะ



รูปที่ 8.8 แสดงแผนภาพ เมื่อกดที่ MENU