

# Line Bot 建立過程簡易說明

## 一、ngrok

使用 ngrok 建立一個臨時的 https 網址(用於接收 Line Bot 傳送過來訊息的網頁須走 https 協定)。首先需要註冊、下載 ngrok，並解壓縮。

打開檔案，並輸入指令(ngrok http 8080)。

```
C:\project\LineBot\ngrok.exe X + v
AUTHOR:
  ngrok - <support@ngrok.com>

COMMANDS:
  config      update or migrate ngrok's configuration file
  http        start an HTTP tunnel
  tcp         start a TCP tunnel
  tunnel      start a tunnel for use with a tunnel-group backen

EXAMPLES:
  ngrok http 80 # secure public URL for port 80 web server
  ngrok http --domain baz.ngrok.dev 8080 # port 8080 available at baz.ngrok.dev
  ngrok tcp 22 # tunnel arbitrary TCP traffic to port 22
  ngrok http 80 --oauth=google --oauth-allow-email=foo@foo.com # secure your app with oauth

Paid Features:
  ngrok http 80 --domain mydomain.com # run ngrok with your own custom domain
  ngrok http 80 --allow-cidr 2600:8c00::a03c:91ee:fe69:9695/32 # run ngrok with IP policy restrictions
  Upgrade your account at https://dashboard.ngrok.com/billing/subscription to access paid features

Upgrade your account at https://dashboard.ngrok.com/billing/subscription to access paid features

Flags:
  -h, --help      help for ngrok

Use "ngrok [command] --help" for more information about a command.

ngrok is a command line application, try typing 'ngrok.exe http 80'
at this terminal prompt to expose port 80.
C:\project\LineBot>ngrok http 8080
```

```
C:\project\LineBot\ngrok.exe X + v
ngrok
Build better APIs with ngrok. Early access: ngrok.com/early-access

Session Status      online
Account             azrael0502@gmail.com (Plan: Free)
Version             3.6.0
Region              Japan (jp)
Latency              -
Web Interface       http://127.0.0.1:4040
Forwarding           https://5e0a-1-160-150-165.ngrok-free.app -> http://localhost:8080

Connections
  ttl    opn    rt1    rt5    p50    p90
    0     0     0.00  0.00  0.00  0.00
```

若是看到這個畫面，表示連線成功，之後需要的是 **Forwarding** 這裡的網址。

## 二、註冊 line business account

建立 channel 之後記下 Basic settings 中的 Channel secret ,以及 Message API 最底下的 Channel access token , 並且把 ngrok 中拿到的網址 ( **Forwarding** ) 後面加上/callback 並 update Message API 底下的 Webhook URL , 也把 Use webhook 打開 , 再到 Auto-reply messages 中 , 把自動回應訊息 disable , 以及啟動 Webhook 。

### Webhook settings

Webhook URL ?

https://47fc-1-160-150-165.ngrok-free.app/callback

- ✓ Don't leave this empty
- ✓ Enter a valid HTTPS URL
- ✓ Enter no more than 500 characters

Update

Cancel

Use webhook ?



Auto-reply  
messages ?

Disabled

Edit ↗

注意 : Webhook URL 因 ngrok 的關係每次啟動專案 URL 會有異動

要注意記得更改。

# 回應設定

可配合帳號的經營目的，設定聊天及自動訊息的回應方式。

---

## 回應功能

聊天



可透過聊天與好友互動。

加入好友的歡迎訊息



當用戶將本帳號加為好友時，可自動傳送訊息內容。

[開啟「加入好友的歡迎訊息」設定畫面](#)

Webhook



當用戶傳送訊息給本帳號或將本帳號加為好友時，從LINE平台傳送Webhook事件至Webhook網址。

[開啟Messaging API的設定畫面](#)

自動回應訊息



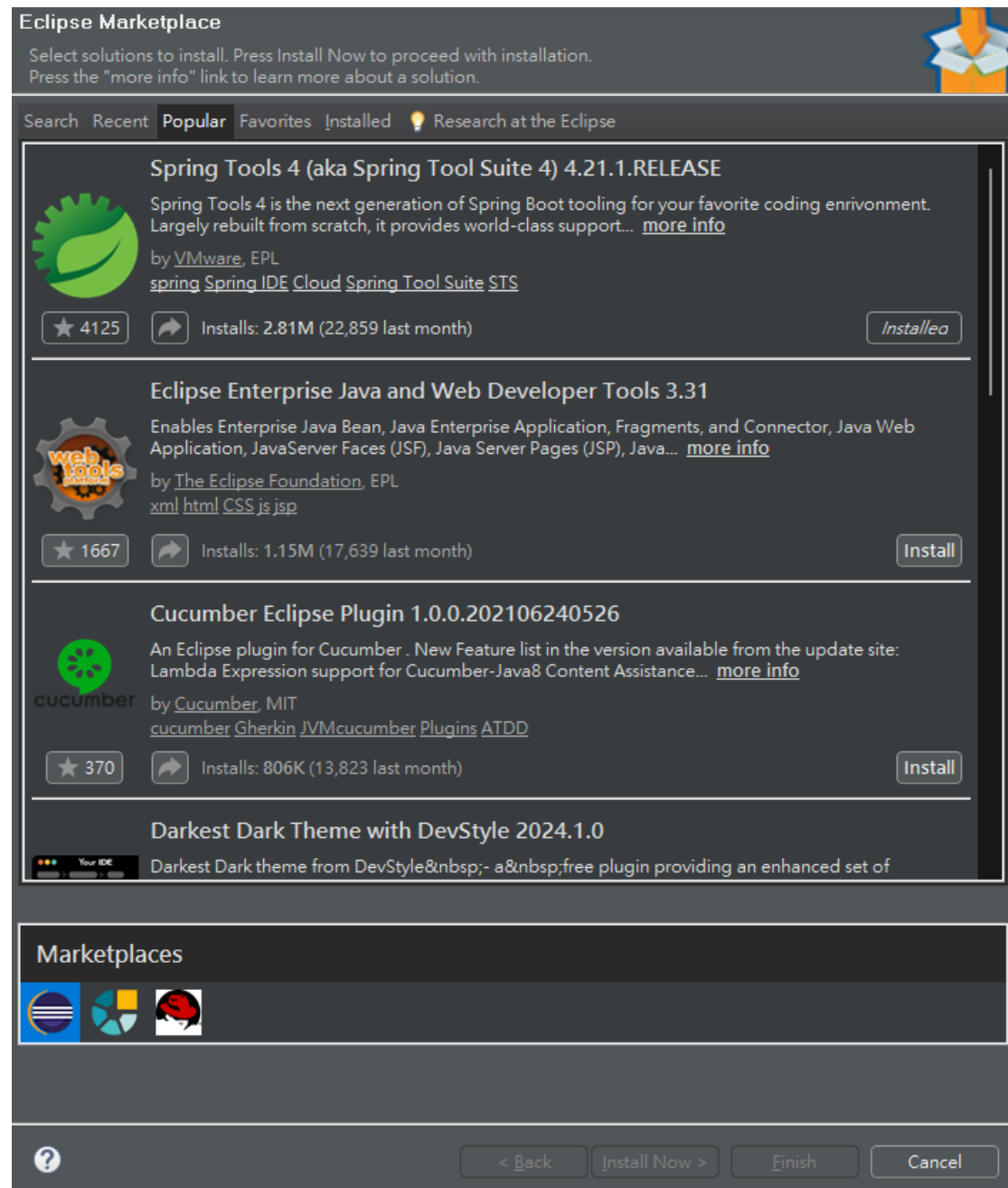
可使用事先設定好的訊息內容進行自動回覆。

[開啟自動回應訊息的設定畫面](#)

### 三、建立 Spring boot 專案

首先必須是要在 Eclipse 內安裝 Spring Tools。

Help -> Eclipse Marketplace 選擇「Popular」標籤就可以找到 Spring Tools

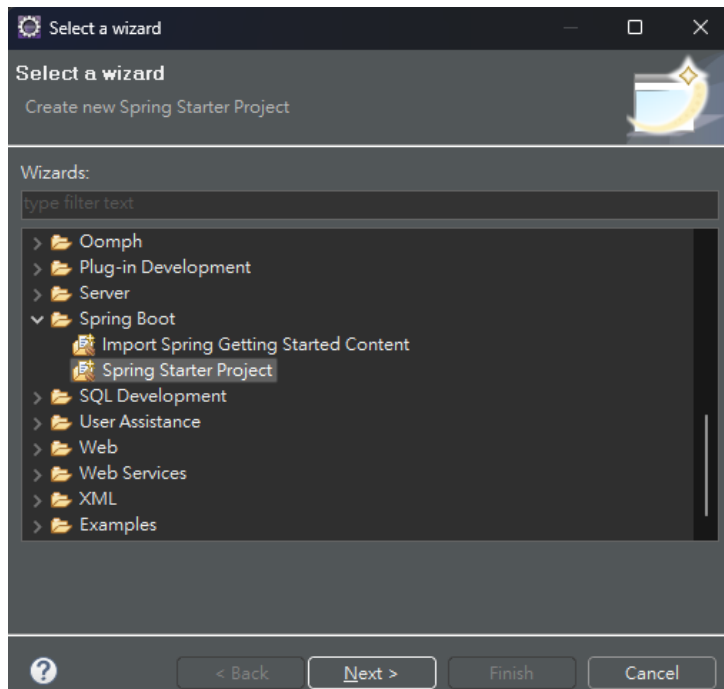


第一步：

打開 Eclipse,

然後在最上方的導覽列上選擇 File > New > Others

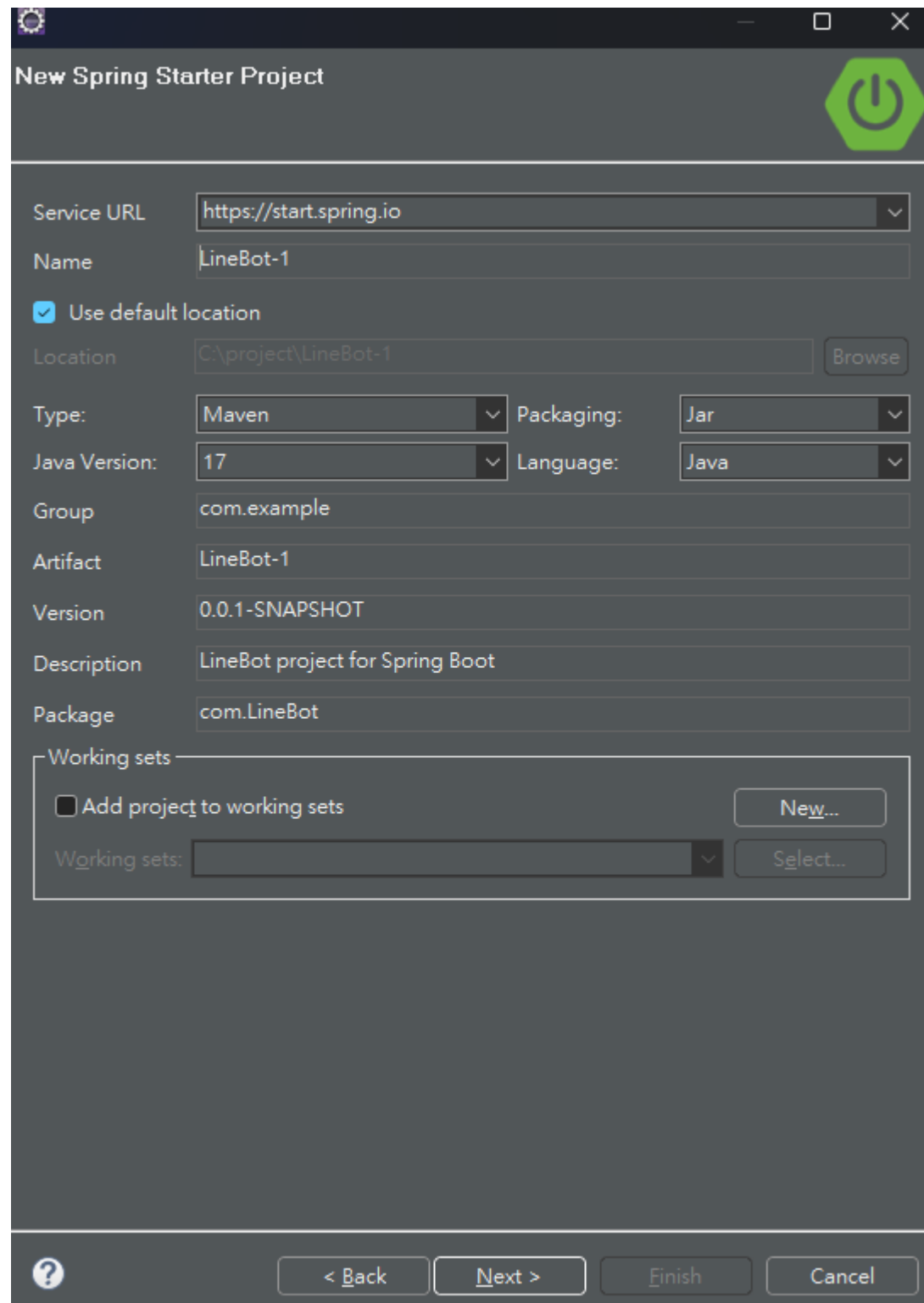
出現以下畫面:



第二步：

在 Wizards 下面點一下 Spring Boot，選 Spring Starter Project，

之後再按一下 Next 就會出現以下畫面：



New Spring Starter Project

Service URL:

Name:

☒ Use default location

Location:

Type:  Packaging:

Java Version:  Language:

Group:

Artifact:

Version:

Description:

Package:

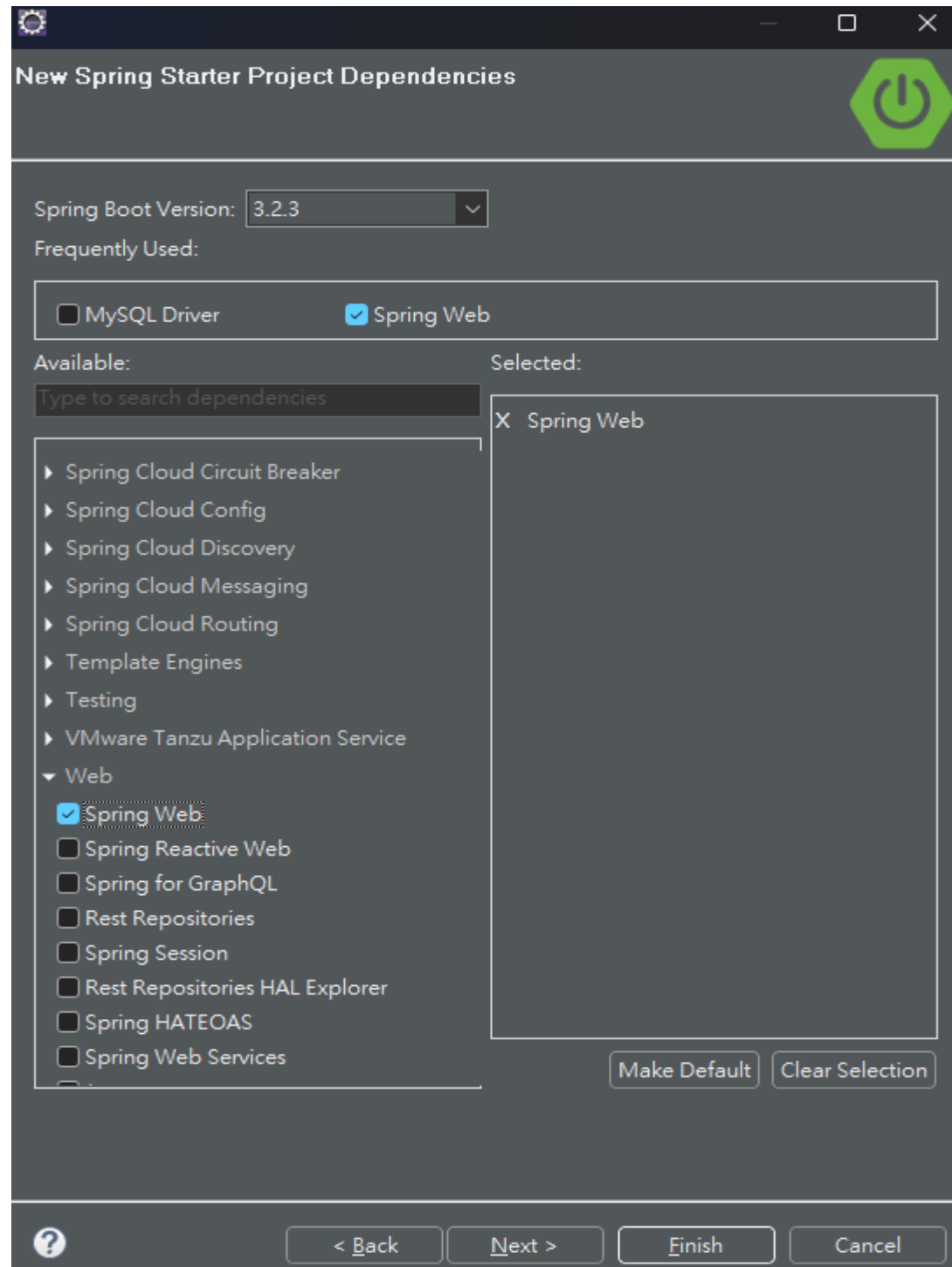
Working sets

☐ Add project to working sets

Working sets:

第三步：

之後再按一下 Next 就會到以下畫面：



只要打開 Web 選 Spring Web 就可以。最後按 Finish 完成建立 Spring Boot 專案。

第四步：

加入 line bot 的 Dependency，pom 檔主要依賴的內容如下：

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-test</artifactId>
  <scope>test</scope>
</dependency>

<dependency>
  <groupId>com.linecorp.bot</groupId>
  <artifactId>line-bot-spring-boot</artifactId>
  <version>4.3.0</version>
</dependency>

<dependency>
  <groupId>com.linecorp.bot</groupId>
  <artifactId>line-bot-model</artifactId>
  <version>4.3.0</version>
</dependency>

<dependency>
  <groupId>com.linecorp.bot</groupId>
  <artifactId>line-bot-api-client</artifactId>
  <version>4.3.0</version>
</dependency>

<dependency>
  <groupId>com.linecorp.bot</groupId>
  <artifactId>line-bot-servlet</artifactId>
  <version>4.3.0</version>
</dependency>
```



第五步：

在 application.properties，把剛剛拿到的 Channel secret 及 Channel access token 貼過來。

```
16
17 server.port=8080
18
19 line.bot.channel-token=Dtk55Jf5u55YCO
20 line.bot.channel-secret=e51c21c7dca93
21 line.bot.handler.enabled=true
22 line.bot.handler.path=/callback
23
```

第六步：

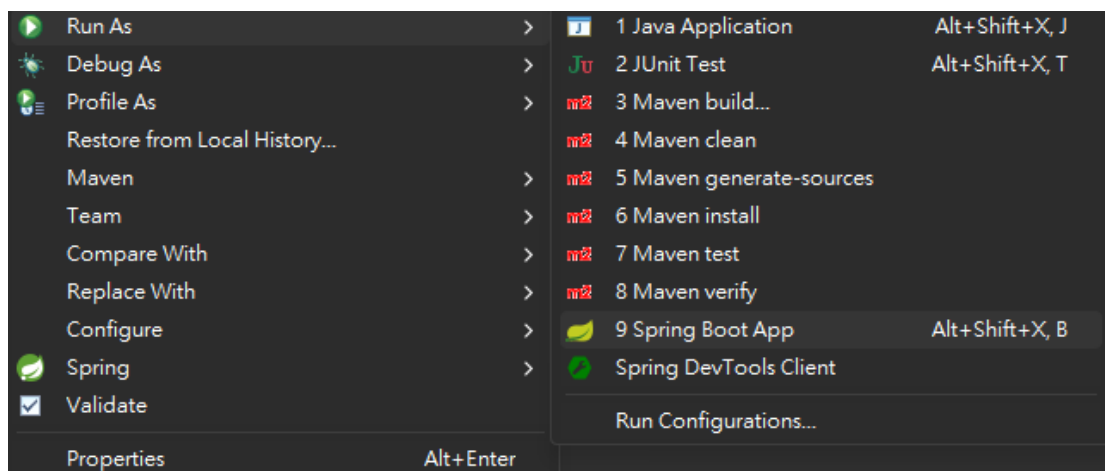
建立一個 messageHandler

```
1 package com.LineBot.controller;
2
3 import org.slf4j.Logger;
4
5
6
7
8
9
10
11
12
13
14
15 @LineMessageHandler
16 public class LinebotHandler {
17
18     private final Logger log = LoggerFactory.getLogger(LinebotHandler.class);
19
20     @RequestMapping
21     public Message handleTextMessageEvent(MessageEvent<TextMessageContent> event) {
22         log.info("event: " + event);
23         final String originalMessageText = event.getMessage().getText();
24         return new TextMessage(originalMessageText);
25     }
26
27     @RequestMapping
28     public void handleDefaultMessageEvent(Event event) {
29         System.out.println("event: " + event);
30     }
31
32 }
33
```

驗證程式：

專案名稱下右鍵點選 Spring Boot App

如下圖：

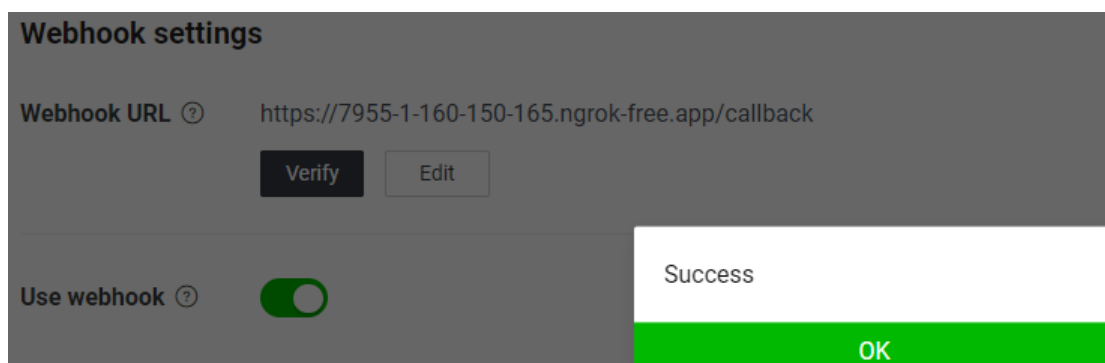


啟動後，確認 line bot 的 api 有作用

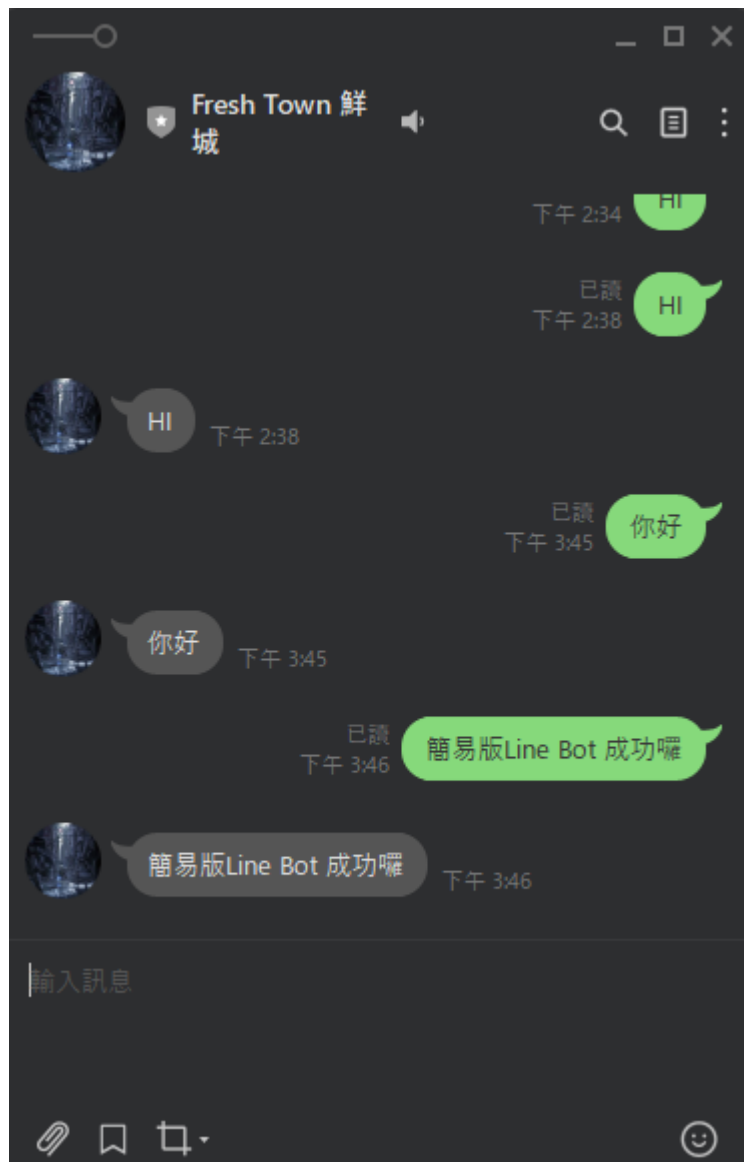
```
2024-03-03 14:38:20.795 INFO 8068 --- [api.line.me/...] com.linecorp.bot.client.wire : x-line-request-id: 9c9b08b3-6c08-461f-alb4-ba36f6912188
2024-03-03 14:38:20.795 INFO 8068 --- [api.line.me/...] com.linecorp.bot.client.wire : x-xss-protection: 1; mode=block
2024-03-03 14:38:20.795 INFO 8068 --- [api.line.me/...] com.linecorp.bot.client.wire : content-length: 212
2024-03-03 14:38:20.795 INFO 8068 --- [api.line.me/...] com.linecorp.bot.client.wire : expires: Sun, 03 Mar 2024 06:38:21 GMT
2024-03-03 14:38:20.795 INFO 8068 --- [api.line.me/...] com.linecorp.bot.client.wire : cache-control: max-age=0, no-cache, no-store
2024-03-03 14:38:20.795 INFO 8068 --- [api.line.me/...] com.linecorp.bot.client.wire : pragma: no-cache
2024-03-03 14:38:20.795 INFO 8068 --- [api.line.me/...] com.linecorp.bot.client.wire : date: Sun, 03 Mar 2024 06:38:21 GMT
2024-03-03 14:38:20.795 INFO 8068 --- [api.line.me/...] com.linecorp.bot.client.wire :
2024-03-03 14:38:20.795 INFO 8068 --- [api.line.me/...] com.linecorp.bot.client.wire : {"sentMessages":[{"id":"497640072612151385","quoteToken":
2024-03-03 14:38:20.796 INFO 8068 --- [api.line.me/...] com.linecorp.bot.client.wire : <-- END HTTP (212-byte body)
2024-03-03 15:38:17.076 INFO 8068 --- [n(65)-127.0.0.1] io.micronaut.runtime.application : Application shutdown requested.
2024-03-03 15:38:17.081 INFO 8068 --- [n(65)-127.0.0.1] o.apache.catalina.core.StandardService : Stopping service [Tomcat]
2024-03-03 15:38:17.082 INFO 8068 --- [n(65)-127.0.0.1] o.a.c.c.C.[Tomcat].[localhost].[/] : Destroying Spring FrameworkServlet 'dispatcherServlet'
```

回到 line developers 的 channel 中，Message API 底下的 Webhook

URL，點擊 Verify，如果彈出 success 的視窗，就代表成功囉！



最後使用 Message API 中的 QR code 進行測試。



以上是 Line Bot 在 Spring Boot 的架構下簡易的建置流程。

參考資料：

<https://hackmd.io/@ohQEG7SsQoeXVwVP2-v06A/spring-boot-line-bot-practice>

<https://ithelp.ithome.com.tw/articles/10214332>

Git hub：

<https://github.com/azrael0502/LineBot>