

使用 lotify + Swagger 建置 可共用的 LINE Notify bot

NiJia

2020/05/27

Chatbot Developer Taiwan
#19 meetup @ Online

Chatbot TW

NiJia Lin

- Backend Developer
- Chatbot Taiwan - Co-organizer
- LINE API Expert
- Farmer / Body builder
- <https://nijialin.com>
- louis70109@gmail.com



Chatbot TW

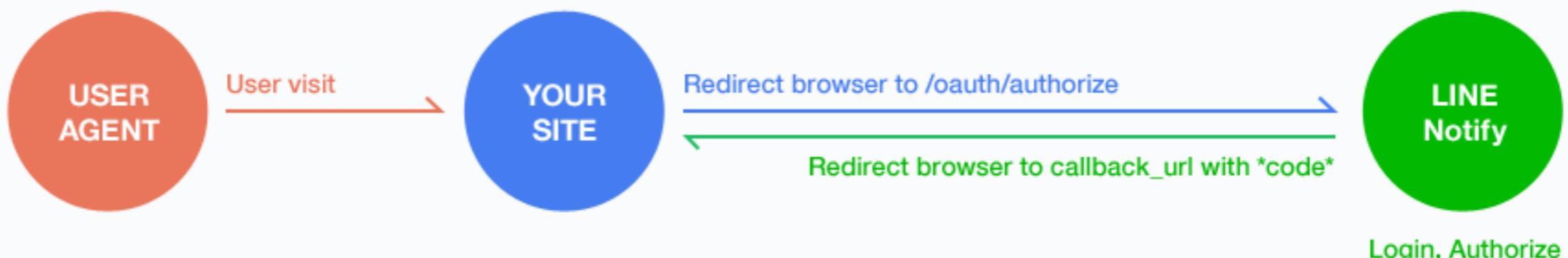
Agenda

- Lotify
 - 介紹
 - 如何使用
 - 快速部署 Heroku
- Client library
 - Swagger 文件
 - Openapi generator
 - JS 使用範例

chatbot TW

LINE Notify 認證流程

User requests integration

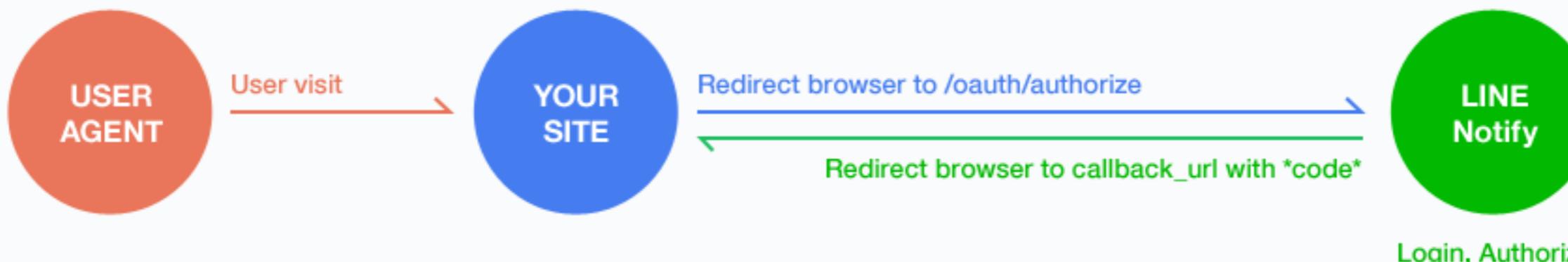


Request access token



LINE Notify 認證流程

User requests integration



Request access token



為什麼要寫

- 它是一個可以讓開發者快速使用 LINE Notify 的套件 
- Pypi 上很多套件功能都只做一半(只有發送功能)
- 每次只要用到 Notify 就重新寫 call API 方法
- 用支援 2.7 的寫法但目前只能支援 3.x

為什麼要寫

- 它是一個可以讓開發者快速使用 LINE Notify 的套件
- Pypi 上很多套件功能都只做一半(只有發送功能)
- 每次只要用到 Notify 就重新寫 call API 方法
- 用支援 2.7 的寫法但目前只能支援 3.x

為什麼要寫

- 它是一個可以讓開發者快速使用 LINE Notify 的套件
- Pypi 上很多套件功能都只做一半(產生網址、換 toekn)
- 每次只要用到 Notify 就重新寫 call API 方法
- 用支援 2.7 的寫法但目前只能支援 3.x

為什麼要寫

- 它是一個可以讓開發者快速使用 LINE Notify 的套件
- Pypi 上很多套件功能都只做一半(只有發送功能)
- 每次只要用到 Notify 就重新寫 call API 方法 😱
- 用支援 2.7 的寫法但目前只能支援 3.x

為什麼要寫

- 它是一個可以讓開發者快速使用 LINE Notify 的套件
- Pypi 上很多套件功能都只做一半(只有發送功能)
- 每次只要用到 Notify 就重新寫 call API 方法
- 用支援 2.7 的寫法但目前只能支援 3.x 😓

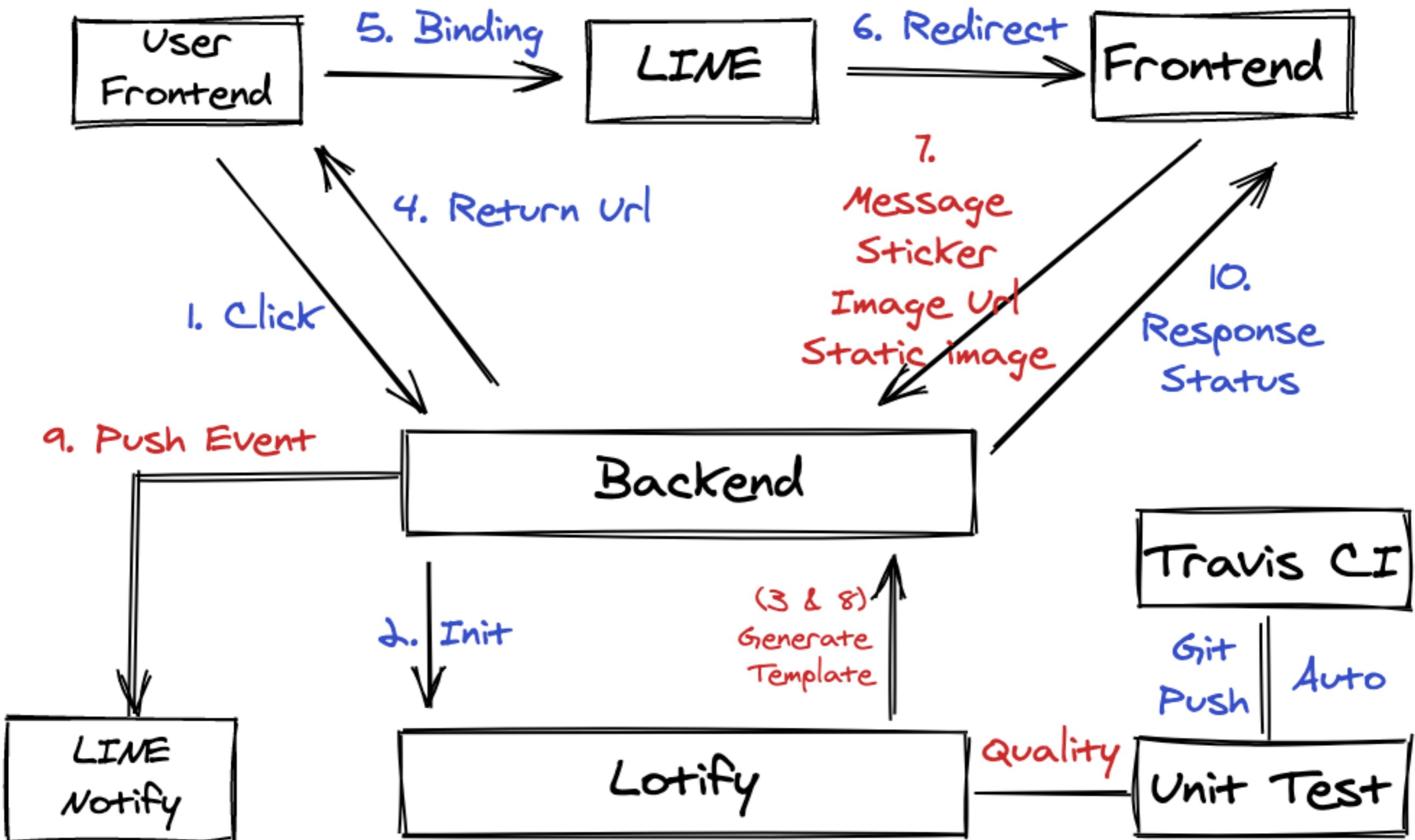


**串接 LINE
Notify API**

使用 Lotify

我加了什麼

- 有測試 ➔ 參考 line-bot-sdk-python 寫法
- 使用文件
- 有範例 ➔ <https://github.com/louis70109/flask-line-notify>
- 還有一鍵部署至 Heroku



老天鵝娛樂

使用 Lotify

就是這麼簡單



如何使用

Chatbot TW

如何使用

```
pip install lotify # 透過 pip 安裝 notify
```

```
from lotify.client import Client
```

```
client = Client(
```

```
    client_id='YOUR_CLIENT_ID',
```

```
    client_secret='YOUR_CLIENT_SECRET',
```

```
    redirect_uri='YOUR_URI'
```

```
)
```

如果有設定環境變數

`LINE_NOTIFY_CLIENT_ID`

`LINE_NOTIFY_CLIENT_SECRET`

`LINE_NOTIFY_REDIRECT_URI`

可以變這樣

```
from lotify.client import Client
```

```
client = Client()
```

```
# 預設讀取環境變數，所以可不用放參數
```



認證步驟

Chatbot TW

前端用法

```
<script>
```

```
function Auth() {  
  
    var URL = 'https://notify-bot.line.me/oauth/authorize?';  
  
    URL += 'response_type=code';  
  
    URL += '&client_id={{CLIENT_ID}}';  
  
    URL += '&redirect_uri={{URI}}';  
  
    URL += '&scope=notify';  
  
    URL += '&state=nostate';  
  
    window.location.href = URL;  
  
}  
  
</script>
```

前端用法

```
<script>
```

```
function Auth() {  
  
    var URL = 'https://notify-bot.line.me/oauth/authorize?';  
  
    URL += 'response_type=code';  
  
    URL += '&client_id={{CLIENT_ID}}';  
  
    URL += '&redirect_uri={{URI}}';  
  
    URL += '&scope=notify';  
  
    URL += '&state=nostate';  
  
    window.location.href = URL;  
  
}  
  
</script>
```

因為要到確認畫面，
所以需要靠前端呼叫 api

取得綁定 Notify 的連結

```
link =  
client.get_auth_link(state='RANDOM_STRING')  
  
print(link)  
  
# https://notify-bot.line.me/oauth/authorize?  
scope=notify&response_type=code&client_id=QxU  
xF.....i51eITH&redirect_uri=http%3A%2F%2Floc  
alhost%3A5000%2Fnotify&state=foo
```

取得綁定 Notify 的連結

```
link =  
client.get_auth_link(state='RANDOM_STRING')
```

```
print(link)
```

```
# https://nc.../authorize?  
scope=notify&response_type=code&client_id=QxU  
xF.....i51eITH&redirect_uri=http%3A%2F%2Floc  
alhost%3A5000%2Fnotify&state=foo
```

Client_id、redirect_uri、state 這些 key
基本上都存後端，
所以讓後端產生 url 紿前端去呼叫

拿回來的 code 換 token

```
access_token =  
client.get_access_token(code='NOTIFY_RESPONSE_CODE')  
  
print(access_token)  
  
# N6g50DiQZk5Xh...25FoFzrs2npkU3z
```

拿回來的 code 換 token

```
access_token =  
client.get_access_token(code='NOTIFY_RESPONSE_CODE')
```

```
print(acce
```

從 LINE 導回來的 code 參數拿來換 access_token 就不需要前端
就可以直接在後端處理

```
# N6g50DiQZk5Xh...25FoFzrs2npkU3z
```



發送系列

Chatbot TW

送文字訊息

```
response =  
client.send_message(access_token='YOUR_ACCESS_  
TOKEN', message='This is notify message')
```

送文字訊息



【鐵人30測試仔】 This is notify message

Chatbot TW

送貼圖訊息

```
response = client.send_message_with_sticker(  
    access_token='YOUR_ACCESS_TOKEN',  
    message='This is notify message',  
    sticker_id=1,  
    sticker_package_id=1)
```

送貼圖訊息

respo

ac

me

sti



【NiJia Notify】 This is notify message



下午 10:52

sticker_package_id=1)

Chatbot TW

<https://github.com/louis70109/lotify#send-message-with-sticker>

送貼圖訊息

respo

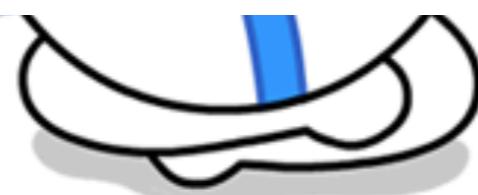


【NiJia Notify】 This is notify message

貼圖包

https://devdocs.line.me/files/sticker_list.pdf

sti



下午 10:52

sticker_package_id=1)

送圖片(檔案)訊息

```
image = client.send_message_with_image_path(  
    access_token='YOUR_ACCESS_TOKEN',  
    message='This is notify message',  
    file=open('./test_image.png', 'rb'))
```

送圖片(檔案)訊息

```
image = client.send_message_with_image_path(  
    access_token='YOUR_ACCESS_TOKEN',  
    message='T (官方只支援 PNG 以及 JPEG 格式)  
    file=open('./test_image.png', 'rb')  
)
```

送圖片(檔案)訊息



【NiJia Notify】 This is notify message

下午 10:57



下午 10:57

儲存 | 另存新檔 | 轉傳 | 貼文串 | Keep

送圖片(網址)訊息

```
image = client.send_message_with_image_url(  
    access_token='YOUR_ACCESS_TOKEN',  
    message='This is notify message',  
    image_thumbnail='https://i.imgur.com/  
RhwZVm.png',  
    image_fullsize='https://i.imgur.com/RhwZVm.png',  
)
```

Chatbot TW

<https://github.com/louis70109/lotify#send-message-with-image-url>

送圖片(網址)訊息

```
image = client.send_message_with_image_url(  
    access_token='YOUR_ACCESS_TOKEN',  
    message='This is a test message.',  
    image_thumbnail='https://i.imgur.com/  
RhwZVm.png',  
    image_fullsize='https://i.imgur.com/RhwZVm.png',  
)
```

雖然 http & https 都吃
還是建議使用 https !

送圖片(網址)訊息

ima



【NiJia Notify】 This is notify message

下午 10:57

a



【鐵人30測試仔】 This is notify message

Rhy

下午 10:57

儲存 | 另存新檔 | 轉傳 | 貼文串 | Keep

) Chatbot TW
<https://github.com/louis70109/lotify#send-message-with-image-url>

註銷 Token

```
revoke =  
client.revoke(access_token='YOUR_ACCESS_TOKEN')
```

註銷 Token



與「鐵人30測試仔」的連動已解除。

re

```
client.revoke(access_token='YOUR_ACCESS_TOKEN')
```

部署至 Heroku

LINE Notify 註冊

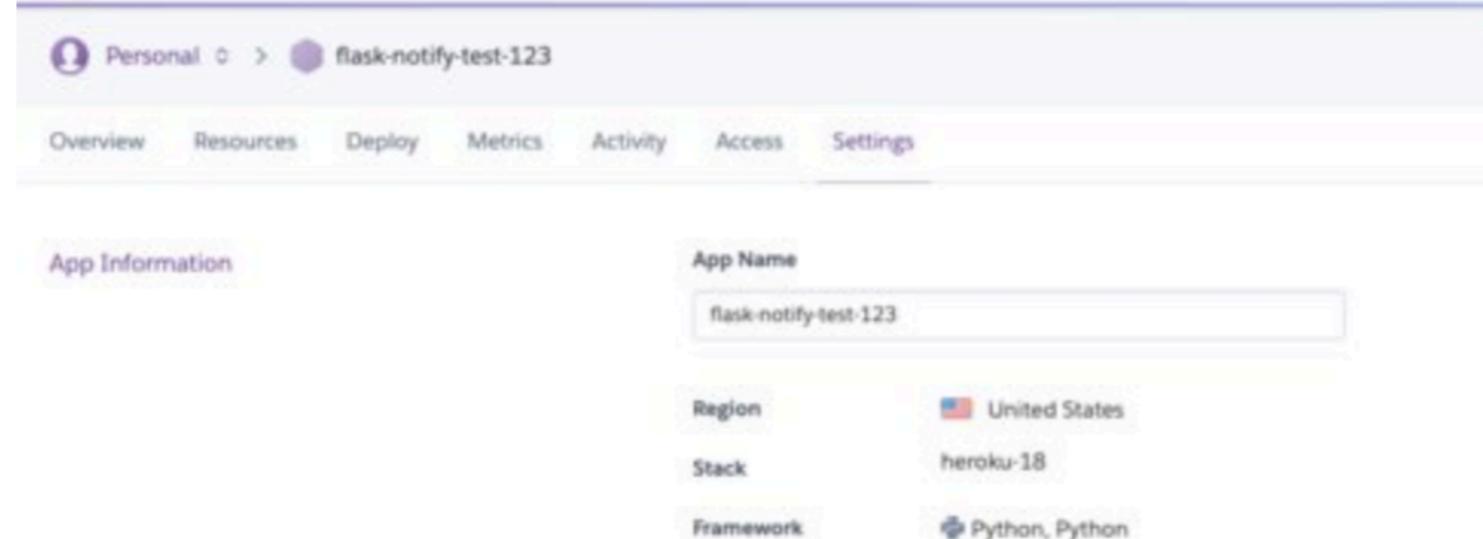
可以參考我之前[鐵人賽的文章](#)。

設定的 Callback Url 為 `http://YOUR_DOMAIN/callback`，本地端測試網址就為 `http://localhost:5000/callback`

一鍵部署

 Deploy to Heroku

按下上面部署按鈕之後需要設定 LINE Notify 所需三個 環境變數

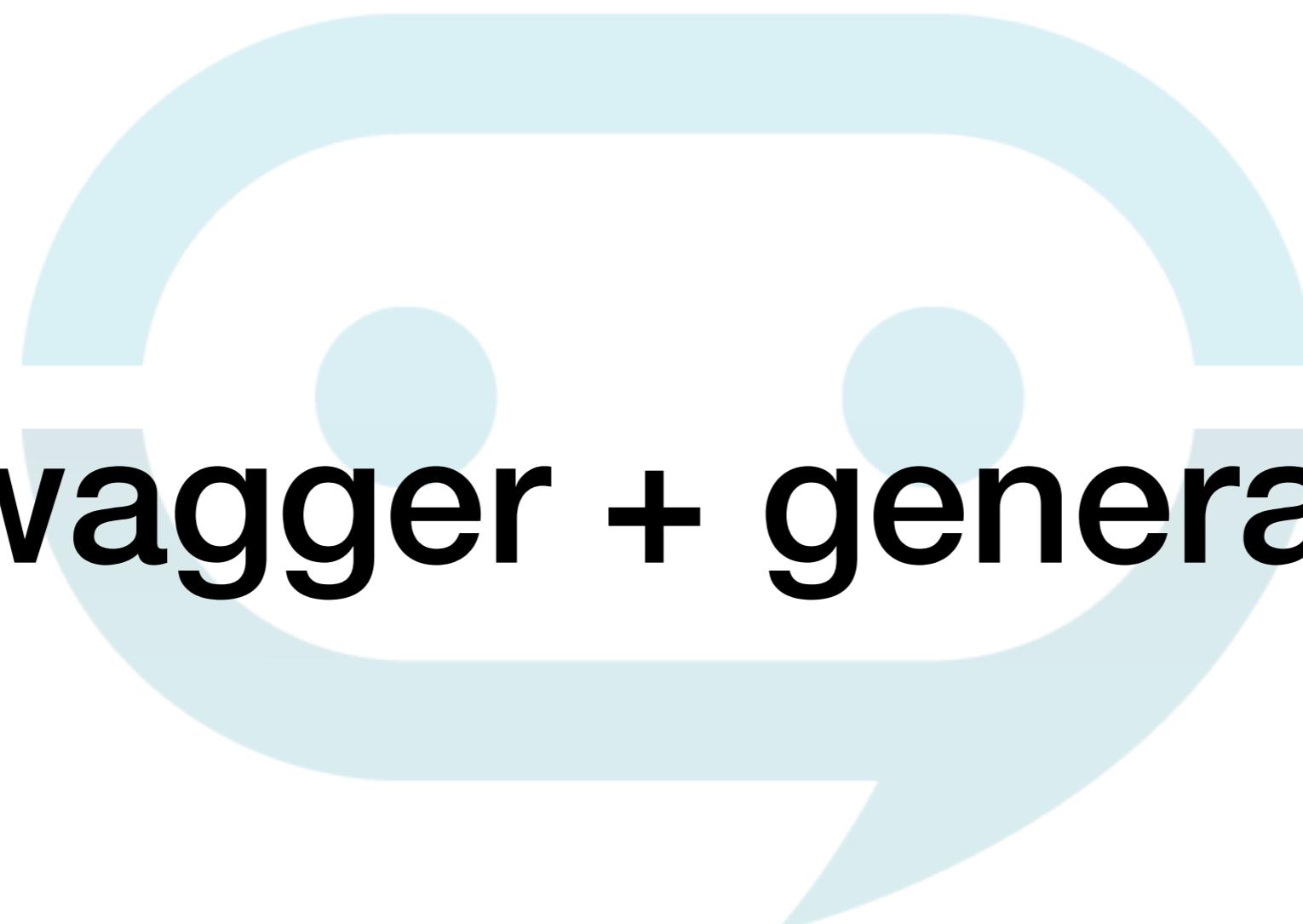


The screenshot shows the Heroku dashboard for the app 'flask-notify-test-123'. The top navigation bar includes 'Personal' and 'flask-notify-test-123'. Below it, there are tabs for 'Overview', 'Resources', 'Deploy', 'Metrics', 'Activity', and 'Settings'. The 'Settings' tab is active. In the 'App Information' section, the 'App Name' is set to 'flask-notify-test-123', 'Region' is 'United States', 'Stack' is 'heroku-18', and 'Framework' is 'Python, Python'.





Chiaogoo RAW



Swagger + generator

Chatbot TW

soerface / flask-restful-swagger-2.0

forked from [rantav/flask-restful-swagger](#)

Watch ▾ 6

Unstar 107

Fork 196

Code Issues 5 Pull requests 2 Actions Projects 0 Wiki Security 0 Insights

A Swagger spec extractor for flask-restful <https://pypi.python.org/pypi/flask-re...>

219 commits

6 branches

0 packages

36 releases

22 contributors

MIT

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾

This branch is 85 commits ahead, 95 commits behind rantav:master.

#28 Compare

soerface Update README.md

Latest commit 4074824 on 19 Aug 2019

example

python3 support

3 years ago

flask_restful_swagger_2

Merge pull request #27 from akabraham/feature/fix-list-and-dict-upstream

11 months ago

tests

Merge branch 'master' into request_parser

3 years ago

.gitignore

Add reparse parsing of query parameters from spec doc.

4 years ago

.travis.yml

Set dist: xenial for python 3.7 support on Travis

11 months ago

LICENSE

Initial commit

7 years ago

MANIFEST

Create a basic setup script

7 years ago

MANIFEST.in

Adding flask_restful_swagger/static to MANIFEST.in

6 years ago

Makefile

Added example code.

4 years ago

原本的已經好幾年沒更新了
因為我習慣使用 flask-restful 套件的朋友可以參考

A Swagger spec extractor for flask-restful <https://pypi.python.org/pypi/flask-re...>

219 commits

6 branches

0 packages

36 releases

22 contributors

MIT

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾

This branch is 85 commits ahead, 95 commits behind rantav:master.

#28 Compare

 soerface Update README.md

✓ Latest commit 4074824 on 19 Aug 2019

 example

python3 support

3 years ago

 flask_restful_swagger_2

Merge pull request #27 from akabraham/feature/fix-list-and-dict-upstream

11 months ago

 tests

Merge branch 'master' into request_parser

3 years ago

 .gitignore

Add reparse parsing of query parameters from spec doc.

4 years ago

 .travis.yml

Set dist: xenial for python 3.7 support on Travis

11 months ago

 LICENSE

Initial commit

7 years ago

 MANIFEST

Create a basic setup script

7 years ago

 MANIFEST.in

Adding flask_restful_swagger/static to MANIFEST.in

6 years ago

 Makefile

Added example code.

4 years ago

特性

- 他是使用 `decorator` 的方式
- 對於不太會 `yaml` 的算是沙漠中的綠洲 (他是 `JSON`)
- 而 `swagger` 也可以解析 `JSON`
- 若不是 `flask-restful` 可以使用 [pyswagger](#)

Chatbot TW

特性

- 他是使用 `decorator` 的方式 (不過挺髒的)
- 對於不太會 `yaml` 的算是沙漠中的綠洲 (他是 `JSON`)
- 而 `swagger` 也可以解析 `JSON`
- 若不是 `flask-restful` 可以使用 [pyswagger](#)

Chatbot TW

特性

- 他是使用 `decorator` 的方式
- 對於不太會 `yaml` 的算是沙漠中的綠洲 (他是 `JSON`)
- 而 `swagger` 也可以解析 `JSON`
- 若不是 `flask-restful` 可以使用 [pyswagger](#)

Chatbot TW

特性

- 他是使用 `decorator` 的方式
- 對於不太會 `yaml` 的算是沙漠中的綠洲 (他是 `JSON`)
- 而 `swagger` 也可以解析 `JSON`
- 若不是 `flask-restful` 可以使用 [pyswagger](#)

Chatbot TW

特性



<https://petstore.swagger.io/v2/swagger.json>

Swagger Petstore 1.0.5

[Base URL: petstore.swagger.io/v2]

<https://petstore.swagger.io/v2/swagger.json>

This is a sample server Petstore server. You can find out more about Swagger at <http://swagger.io> authorization filters.

[Terms of service](#)

[Contact the developer](#)

[Apache 2.0](#)

[Find out more about Swagger](#)

特性

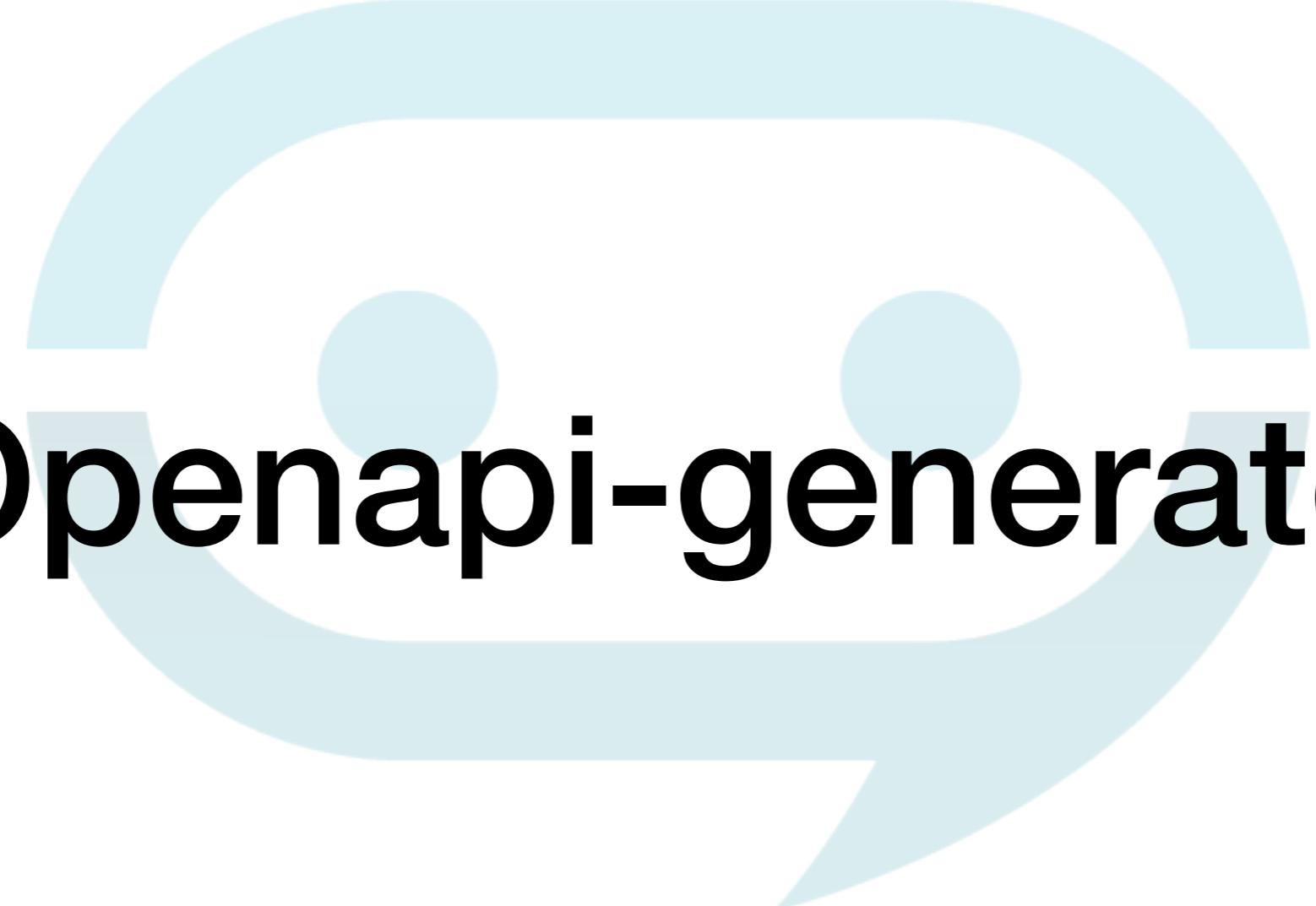
- 他是使用 `decorator` 的方式
- 對於不太會 `yaml` 的算是沙漠中的綠洲 (他是 `JSON`)
- 而 `swagger` 也可以解析 `JSON`
- 若不是 `flask-restful` 可以使用 [pyswagger](#)

Chatbot TW



Cha'

TW



Openapi-generator

Chatbot TW

[Code](#) [Issues 1,424](#) [Pull requests 206](#) [Actions](#) [Projects 6](#) [Wiki](#) [Security 0](#) [Insights](#)

OpenAPI Generator allows generation of API client libraries (SDK generation), server stubs, documentation and configuration automatically given an OpenAPI Spec (v2, v3) <https://openapi-generator.tech>

[rest-api](#) [rest-client](#) [sdk](#) [generator](#) [restful-api](#) [api](#) [api-client](#) [api-server](#) [openapi3](#) [openapi](#) [rest](#) [openapi-generator](#)[14,122 commits](#)[61 branches](#)[0 packages](#)[55 releases](#)[1,466 contributors](#)[Apache-2.0](#)

Branch: master ▾

[New pull request](#)[Create new file](#)[Upload files](#)[Find file](#)[Clone or download](#) ▾

wing328 add pub to struct in single parameter (#6381)

Latest commit 2ec0754 9 minutes ago

[.github](#)

Update sonar.yml

[.mvn/wrapper](#)

#7093 - Add maven wrapper (#7356)

[CI](#)

Mark swift4 generator as deprecated (#6311)

[bin](#)

[Kotlin][Client] Added supporting Retrofit2(RxJava/RxJava2/Corouti

[docs](#)

[Kotlin][Client] Added supporting Retrofit2(RxJava/RxJava2/Corouti

[modules](#)

add pub to struct in single parameter (#6381)

[samples](#)

replacing caTools dependency with base64enc (#6349)

[scripts](#)

Making completion work with openapi-generator-cli or openapi-generator (

掃他進去看看



16 months ago

支援語言

	Languages/Frameworks
API clients	ActionScript, Ada, Apex, Bash, C, C# (.net 2.0, 3.5 or later, .NET Standard 1.3 - 2.0, .NET Core 2.0), C++ (cpp-restsdk, Qt5, Tizen), Clojure, Dart (1.x, 2.x), Elixir, Elm, Eiffel, Erlang, Go, Groovy, Haskell (http-client, Servant), Java (Jersey1.x, Jersey2.x, OkHttp, Retrofit1.x, Retrofit2.x, Feign, RestTemplate, RESTEasy, Vertx, Google API Client Library for Java, Rest-assured, Spring 5 Web Client, MicroProfile Rest Client), k6, Kotlin, Lua, Nim, Node.js/JavaScript (ES5, ES6, AngularJS with Google Closure Compiler annotations, Flow types, Apollo GraphQL DataStore), Objective-C, OCaml, Perl, PHP, PowerShell, Python, R, Ruby, Rust (rust, rust-server), Scala (akka, http4s, scalaz, sttp, swagger-async-httpclient), Swift (2.x, 3.x, 4.x, 5.x), Typescript (AngularJS, Angular (2.x - 8.x), Aurelia, Axios, Fetch, Inversify, jQuery, Node, Rxjs)
Server stubs	Ada, C# (ASP.NET Core, NancyFx), C++ (Pistache, Restbed, Qt5 QHTTPEngine), Erlang, F# (Giraffe), Go (net/http, Gin), Haskell (Servant), Java (MSF4J, Spring, Undertow, JAX-RS: CDI, CXF, Inflector, Jersey, RestEasy, Play Framework, PKMST, Vert.x), Kotlin (Spring Boot, Ktor, Vertx), PHP (Laravel, Lumen, Slim, Silex, Symfony, Zend Expressive), Python (Flask), NodeJS, Ruby (Sinatra, Rails5), Rust (rust-server), Scala (Akka, Finch, Lagom, Play, Scalatra)
API documentation generators	HTML, Confluence Wiki, Asciidoc
Configuration files	Apache2
Others	GraphQL, JMeter, MySQL Schema, Protocol Buffer

Openapi-generator

- Swagger document 寫好就沒問題
- 可以針對指定環境加參數設定 (等等有範例)
- 就是使用套件，而它已經包裝完了

Chatbot TW

api

0.0.1

[Base URL: localhost:5000 /]
<http://localhost:5000/api/swagger.json>

Schemes

HTTP



Text

POST [/notify/file](#)

POST [/notify/revoke](#)

POST [/notify/send](#)

POST [/notify/sticker](#)

POST [/notify/url](#)

Models

[RevokeRequestBody](#) >

[NotifyResponse](#) >

Openapi-generator

- Swagger document 寫好就沒問題
- 可以針對指定環境加參數設定 (等等有範例)
- 就是使用套件，而它已經包裝完了

Chatbot TW

Openapi-generator

- Swagger document 寫好就沒問題
- 可以針對指定環境加參數設定 (等等有範例)

Openapi-gererator 安裝支援 mvn, docker, brew, npm

這裡我就用 npm demo

(不管用哪個它一定要安裝 JDK)

Chatbot TW

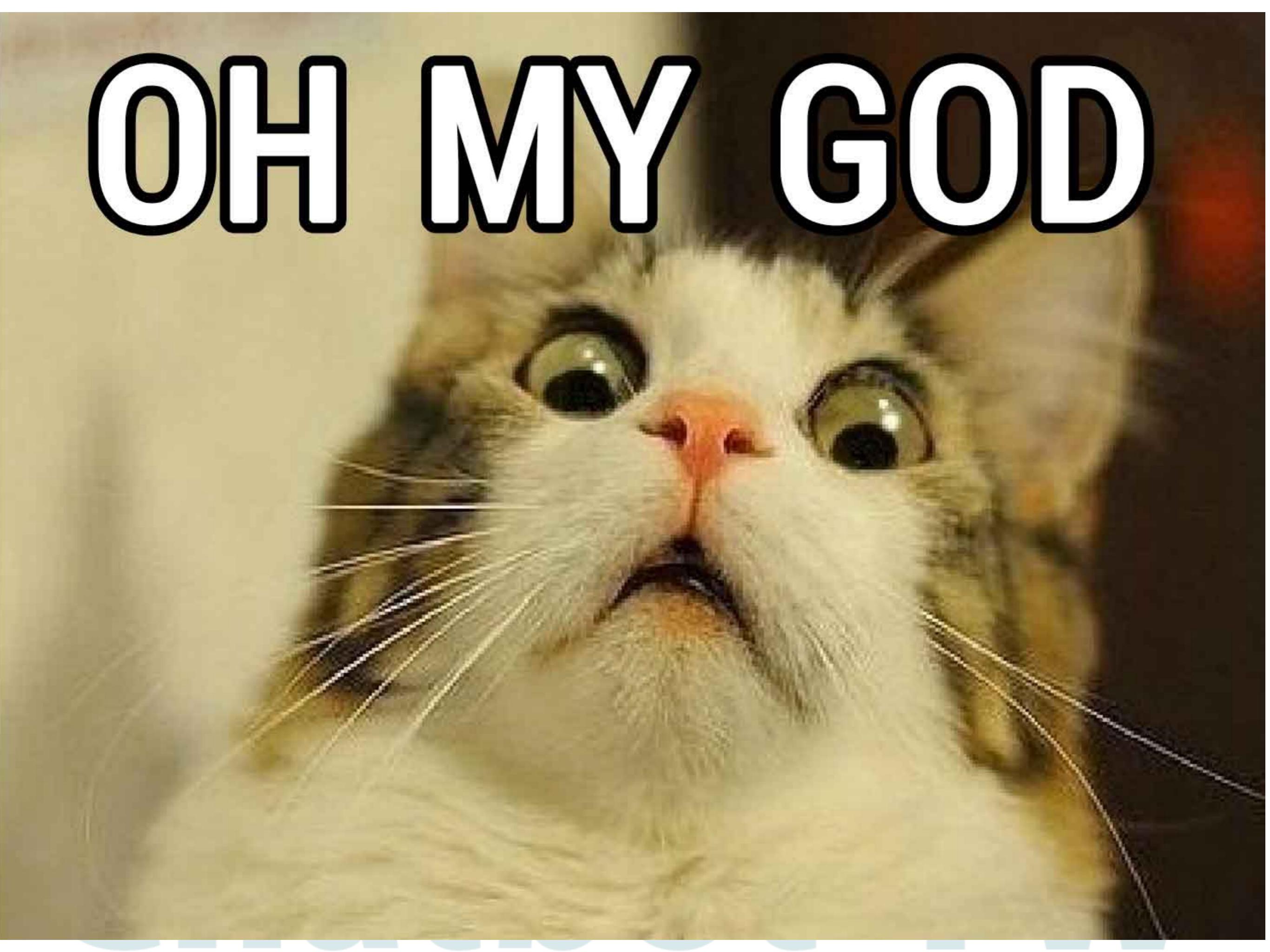
Openapi-generator

- Swagger document 寫好就沒問題
- 可以針對指定環境加參數設定 (等等有範例)

Openapi-gererator 安裝支援 mvn, docker, brew, npm
這裡我就用 npm demo
(不管用哪個它一定要安裝 JDK)

No Java runtime present, requesting install.

OH MY GOD



Openapi-generator

```
[Programs] openapi-generator generate -i http://localhost:5000/api/swagger.json -g javascript -o api  
--additional-properties=useES6=false  
[main] INFO o.o.codegen.DefaultGenerator - Generating with dryRun=false  
[main] WARN o.o.c.ignore.CodegenIgnoreProcessor - Output directory does not exist, or is inaccessible.  
. No file (.openapi-generator-ignore) will be evaluated.  
[main] INFO o.o.codegen.DefaultGenerator - OpenAPI Generator: javascript (client)  
[main] INFO o.o.codegen.DefaultGenerator - Generator 'javascript' is considered stable.  
[main] WARN o.o.codegen.utils.ModelUtils - Multiple schemas found in the OAS 'content' section, return-  
ning only the first one (*/*)  
[main] WARN o.o.codegen.utils.ModelUtils - Multiple schemas found in the OAS 'content' section, return-  
ning only the first one (*/*)  
[main] WARN o.o.codegen.utils.ModelUtils - Multiple schemas found in the OAS 'content' section, return-  
ning only the first one (*/*)  
[main] WARN o.o.codegen.utils.ModelUtils - Multiple schemas found in the OAS 'content' section, return-  
ning only the first one (*/*)  
[main] WARN o.o.codegen.utils.ModelUtils - Multiple schemas found in the OAS 'content' section, return-  
ning only the first one (*/*)  
[main] WARN o.o.codegen.utils.ModelUtils - Multiple schemas found in the OAS 'content' section, return-  
ning only the first one (*/*)
```

Openapi-generator

額外參數

Input url

語言

輸出

```
[Programs] openapi-generator generate -i http://localhost:5000/api/swagger.json -g javascript -o api  
--additional-properties=useES6=false
```

[main] INFO o.o.codegen.DefaultGenerator - Generating with dryRun=false

[main] WARN o.o.c.ignore.CodegenIgnoreProcessor - Output directory does not exist, or is inaccessible.
. No file (.openapi-generator-ignore) will be evaluated.

[main] INFO o.o.codegen.DefaultGenerator - OpenAPI Generator: javascript (client)

[main] INFO o.o.codegen.DefaultGenerator - Generator 'javascript' is considered stable.

[main] WARN o.o.codegen.utils.ModelUtils - Multiple schemas found in the OAS 'content' section, returning only the first one (*/*)

[main] WARN o.o.codegen.utils.ModelUtils - Multiple schemas found in the OAS 'content' section, returning only the first one (*/*)

[main] WARN o.o.codegen.utils.ModelUtils - Multiple schemas found in the OAS 'content' section, returning only the first one (*/*)

[main] WARN o.o.codegen.utils.ModelUtils - Multiple schemas found in the OAS 'content' section, returning only the first one (*/*)

[main] WARN o.o.codegen.utils.ModelUtils - Multiple schemas found in the OAS 'content' section, returning only the first one (*/*)

[main] WARN o.o.codegen.utils.ModelUtils - Multiple schemas found in the OAS 'content' section, returning only the first one (*/*)

chatbotTW

[Code](#)[Issues 0](#)[Pull requests 0](#)[Actions](#)[Projects 0](#)[Wiki](#)[Security 0](#)[Insights](#)[Settings](#)

No description, website, or topics provided.

我傳到 Github 做範例

[Edit](#)[Manage topics](#)[1 commit](#)[1 branch](#)[0 packages](#)[0 releases](#)[1 contributor](#)

Branch: master ▾

[New pull request](#)[Create new file](#)[Upload files](#)[Find file](#)[Clone or download ▾](#)

 louis70109 init	✓ Latest commit e9e7cf5 5 minutes ago
.openapi-generator	init 5 minutes ago
docs	init 5 minutes ago
src	init 5 minutes ago
test	init 5 minutes ago
.babelrc	init 5 minutes ago
.openapi-generator-ignore	init 5 minutes ago
.travis.yml	init 5 minutes ago
README.md	init 5 minutes ago
git_push.sh	init 5 minutes ago
mocha.opts	init 5 minutes ago
package.json	init 5 minutes ago

 Code

 Issues 0

 Pull requests

Branch: master ▾ [api / docs /](#)

 louis70109 update

..

 [ImageApi.md](#)

 [ImageUrlRequestBody.md](#)

 [NotifyResponse.md](#)

 [RevokeApi.md](#)

 [RevokeRequestBody.md](#)

 [StickerApi.md](#)

 [StickerRequestBody.md](#)

 [TextApi.md](#)

 [TextRequestBody.md](#)

Api 產生後他也很好心的產生說明文件
看這裡的內容就知道提供的人是不是好人了

所有的範例跟說明都在這裡面

bot TW

Openapi-generator

- Swagger document 寫好就沒問題
- 可以針對指定環境加參數設定 (等等有範例)
- 就是使用套件，而它已經包裝完了

Chatbot TW

以 Bottender 為 JS 範例

- `npx create-bottender-app my-app`
- `cd my-app`
- `npm install https://github.com/{帳號}/api.git --save`

louis70109 / api

Code Issues 0 Pull requests

Branch: master ▾ api / docs /

 louis70109 update

..

[ImageApi.md](#)

[ImageUrlRequestBody.md](#)

[NotifyResponse.md](#)

[RevokeApi.md](#)

[RevokeRequestBody.md](#)

[StickerApi.md](#)

[StickerRequestBody.md](#)

[TextApi.md](#)

[TextRequestBody.md](#)

主要看的地方在這邊

Api.StickerApi

All URIs are relative to <http://localhost:5000>

Method	HTTP request	Description
sendSticker	POST /notify/sticker	

sendSticker

NotifyResponse sendSticker(body)
Send LINE Notify text message with sticker

直接複製範例 code 來用 😊

Example

```
var Api = require('api');

var apiInstance = new Api.StickerApi();
var body = new Api.StickerRequestBody(); // StickerRequestBody | Send LINE Notify text message with sticker
var callback = function(error, data, response) {
  if (error) {
    console.error(error);
  } else {
    console.log('API called successfully. Returned data: ' + data);
  }
};
apiInstance.sendSticker(body, callback);
```

Parameters

Name	Type	Description	Notes
body	StickerRequestBody	Send LINE Notify text message with sticker	

master/docs/StickerApi.md#apistickerapi

StickerRequestBody

index.js — my-app

js index.js ×

src > js index.js > <unknown> > App > [?] callback

```
1 var Api = require('api');
2
3 Complexity is 3 Everything is cool!
4 module.exports = async function App(context) {
5     var apiInstance = new Api.StickerApi();
6     var body = new Api.StickerRequestBody(); // StickerRequestBody
7     body.token = 'p82kdl0zw98CHh9jb5dhFeXkJQVjmuI60zFurjun0Xr';
8     body.message = 'test openapi';
9     body.sticker_id = 1;
10    body.package_id = 1;
11    var callback = function (error, data, response) {
12        if (error) {
13            console.error(error);
14        }
15        apiInstance.sendSticker(body, callback);
16        await context.sendText('Send sticker successfully');
17    };
18}
```

直接把提供的文件 code 複製過來改寫

④ 2 △ 0 ⓘ 2 ⌂ Live Share {..}:0 ✓ javascript | ✓ index.js UTF-8 LF Babel JavaScript ESLint Prettier ⌂ ⌂

結果

```
App has started
You > hello
hello
Bot > Send sticker successfully
You >
```



【NiJia Notify】 test openapi



下午 2:06

chatbot



**這類的 Open Source
星星我一律按爆**

(但是記得按奇數下)



謝謝大家！！

Q & A 嘅

Chatbot TW