

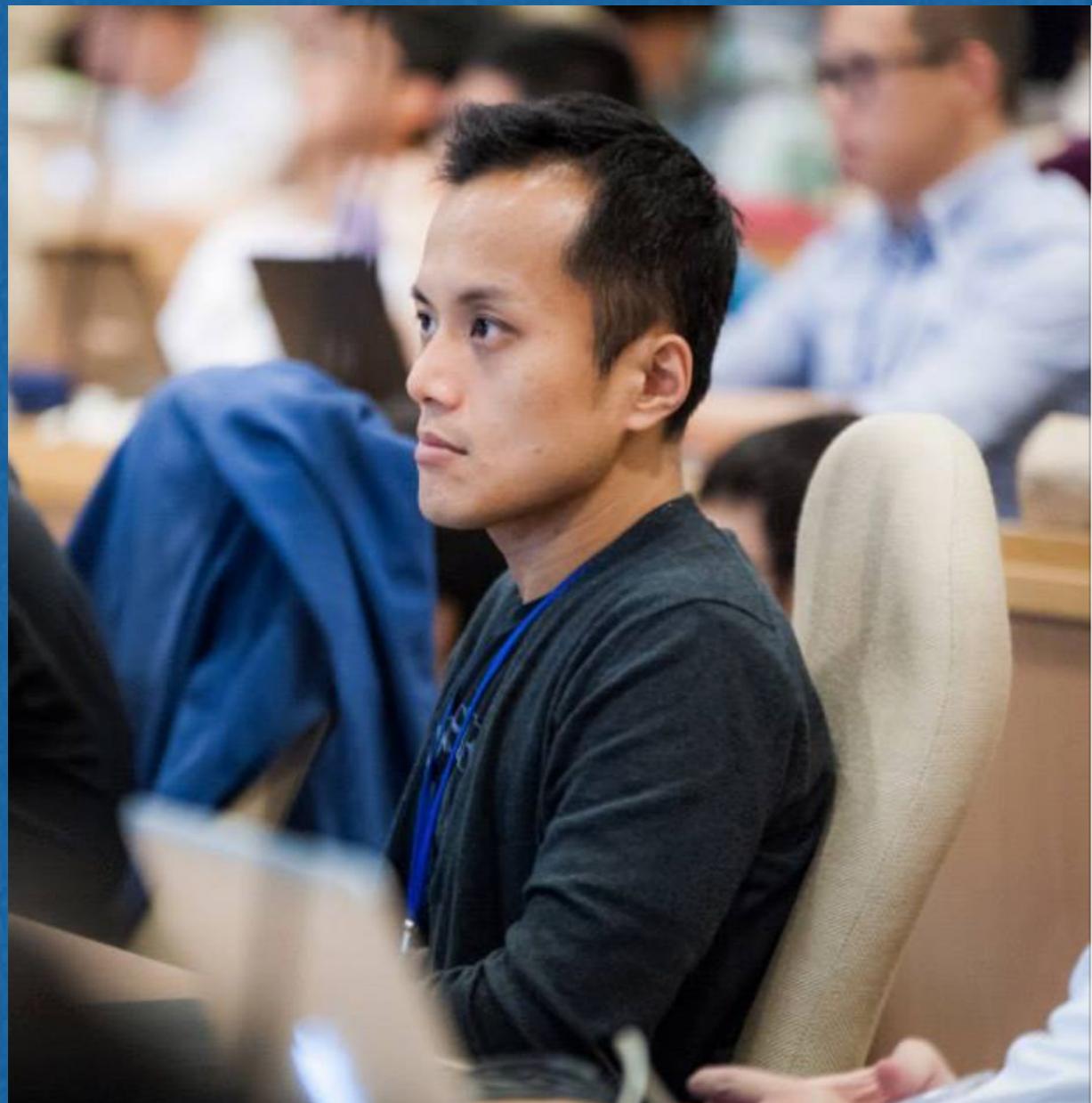


Golang Taipei Gathering #16

Using Golang to build smart IM Bot

Evan Lin @LinkerNetworks

About Me



- Cloud Architect @ Linker Networks
- Top 10 Taiwan Golang open source contributor (github award)
- Developer, Curator, Blogger

 **CodeTengu**
 **yowuReport**
有物報告



How to choose programming language

- Easy to Write
- Fast in Run and Compile
- Powerful Concurrency
- Powerful Built-in Toolchain



Pokemon GO

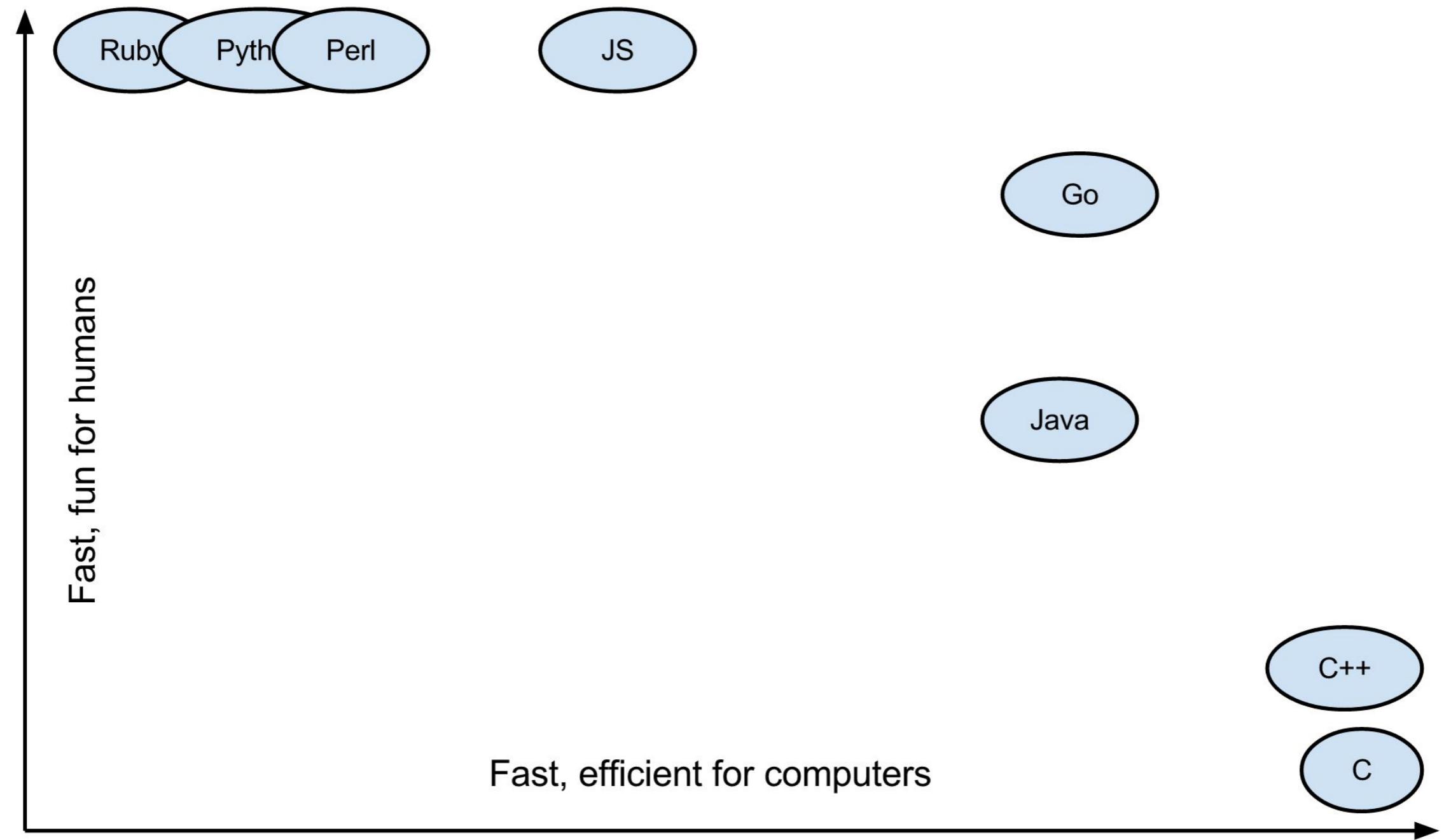


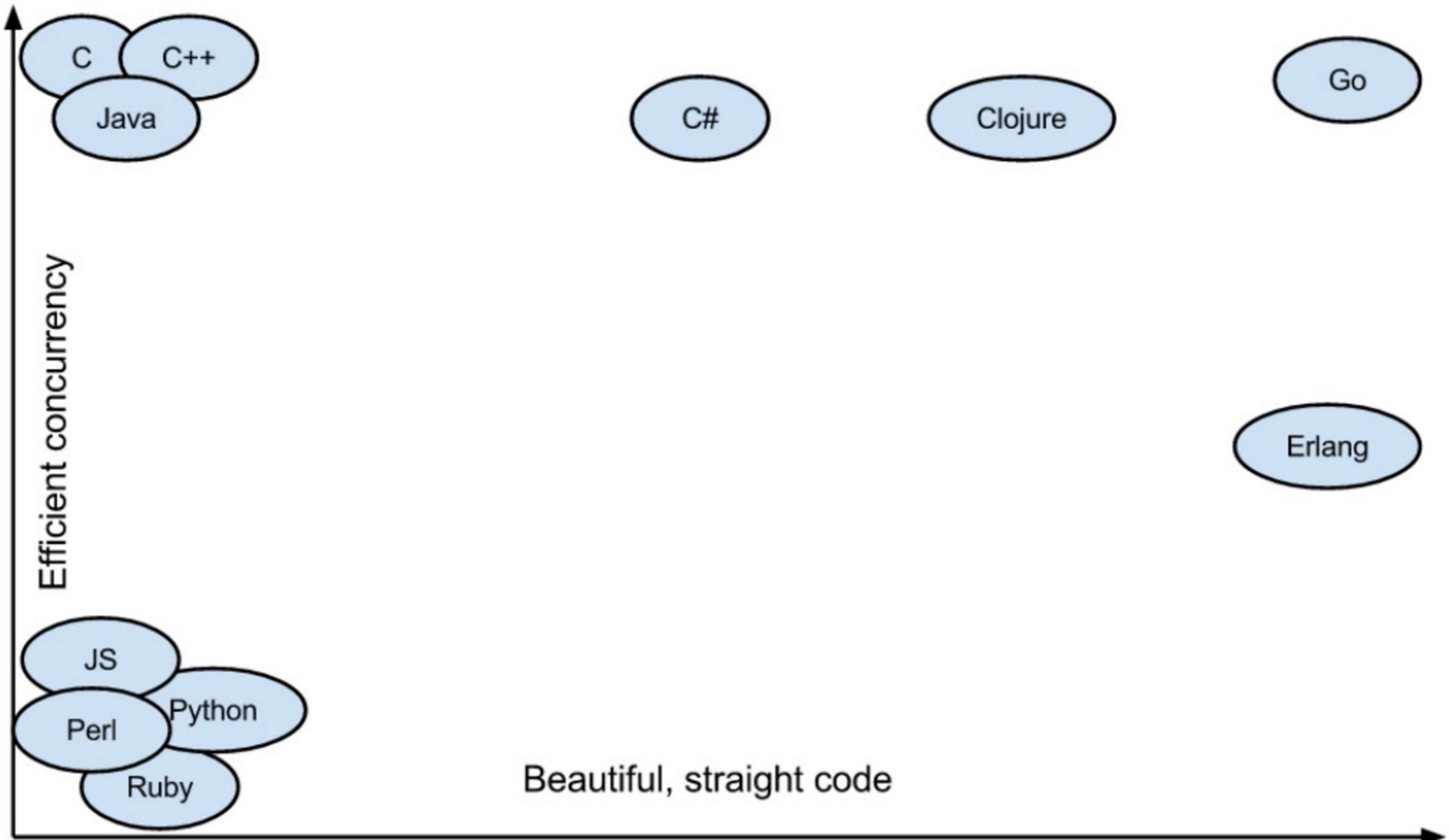
<p>CP45 上班東西趕不出來 HP 17 / 17 みず タイプ 14.2 kg 重さ 0.75 m 高さ</p>	<p>CP45 回家路上抓不到稀有怪 HP 17 / 17 みず タイプ 14.2 kg 重さ 0.75 m 高さ</p>
<p>CP45 乾脆來寫 Go HP 17 / 17 みず タイプ 14.2 kg 重さ 0.75 m 高さ</p>	<p>CP45 又快又棒呢! HP 17 / 17 みず タイプ 14.2 kg 重さ 0.75 m 高さ</p>



Go is Fast

"Go: 90% Perfect, 100% of the time" -bradfitz, 2014







gofmt / goimport



I mean I do not get why anyone
would use spaces over tabs



Before we look code

ISSUE 46 不能信任那些 TERMINAL 或編輯器用白底的人 *June 6th 2016*

HELLO WORLD

CodeTengu Weekly 碼天狗週刊



Before we look code (cont)

ISSUE 19 我早年都是用 VIM 寫程式，也說不上特別喜歡，主要是當時還不知道怎麼退出 VIM

December 7th 2015

HELLO WORLD

CodeTengu Weekly 碼天狗週刊

```
1 package main
2
3 import (
4     "fmt"
5     "log"
6     "time"
7 )
8
9 func sum(a []int, c chan int) { █
10    fmt.Println("Arra is =", a)
11    sum := 0
12    for _, v := range a { sum += v }
13    c <- sum // send sum to c
14 }
15
16 func fabnacci(sum chan int, quit chan int) (ret int) █
17 {
18     x, y := 0, 1
19     for {
20         select {
21             case sum <- x: x, y = y, x+y
22             case <-quit:
23                 return
24             default:
25                 fmt.Println("timer call")
26                 time.Sleep(50 * time.Millisecond) }
27     } }
```

```
1 package main
2
3 import (
4     "fmt"
5     "log"
6     "time"
7 )
8
9 func sum(a []int, c chan int) {
10    fmt.Println("Arra is =", a)
11    sum := 0
12    for _, v := range a { sum += v }
13    c <- sum // send sum to c
14 }
15
16 func fabnacci(sum chan int, quit chan int) (ret int) {
17 {
18    x, y := 0, 1
19    for {
20        select {
21            case sum <- x: x, y = y, x+y
22            case <-quit:
23                return
24            default:
25                fmt.Println("timer call")
26                time.Sleep(50 * time.Millisecond) }
27    } }
```

```
1 package main
2
3 import (
4     "fmt"
5     "time"
6 )
7
8 func sum(a []int, c chan int) {
9     fmt.Println("Arra is =", a)
10    sum := 0
11    for _, v := range a {
12        sum += v
13    }
14    c <- sum // send sum to c
15 }
16
17 func fabnacci(sum chan int, quit chan int) (ret int) {
18     x, y := 0, 1
19     for {
20         select {
21             case sum <- x:
22                 x, y = y, x+y
23             case <-quit:
24                 return
25             default:
26                 fmt.Println("timer call")
27                 time.Sleep(50 * time.Millisecond)
28         }
29     }
30 }
```

After file save...



GoMobile

Original Work Flow



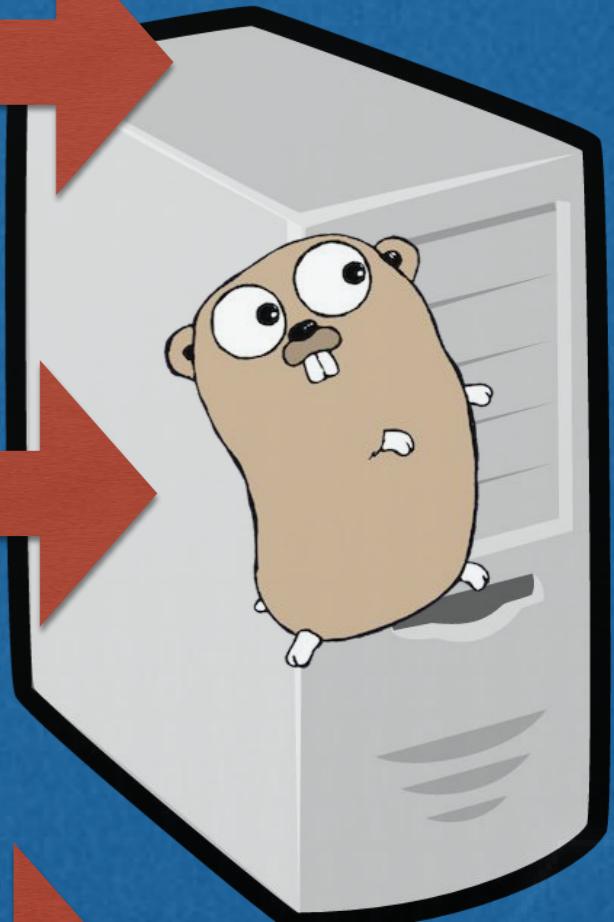
Sometime Failed



Failed on Some Model

Golang
Sample Console

Work





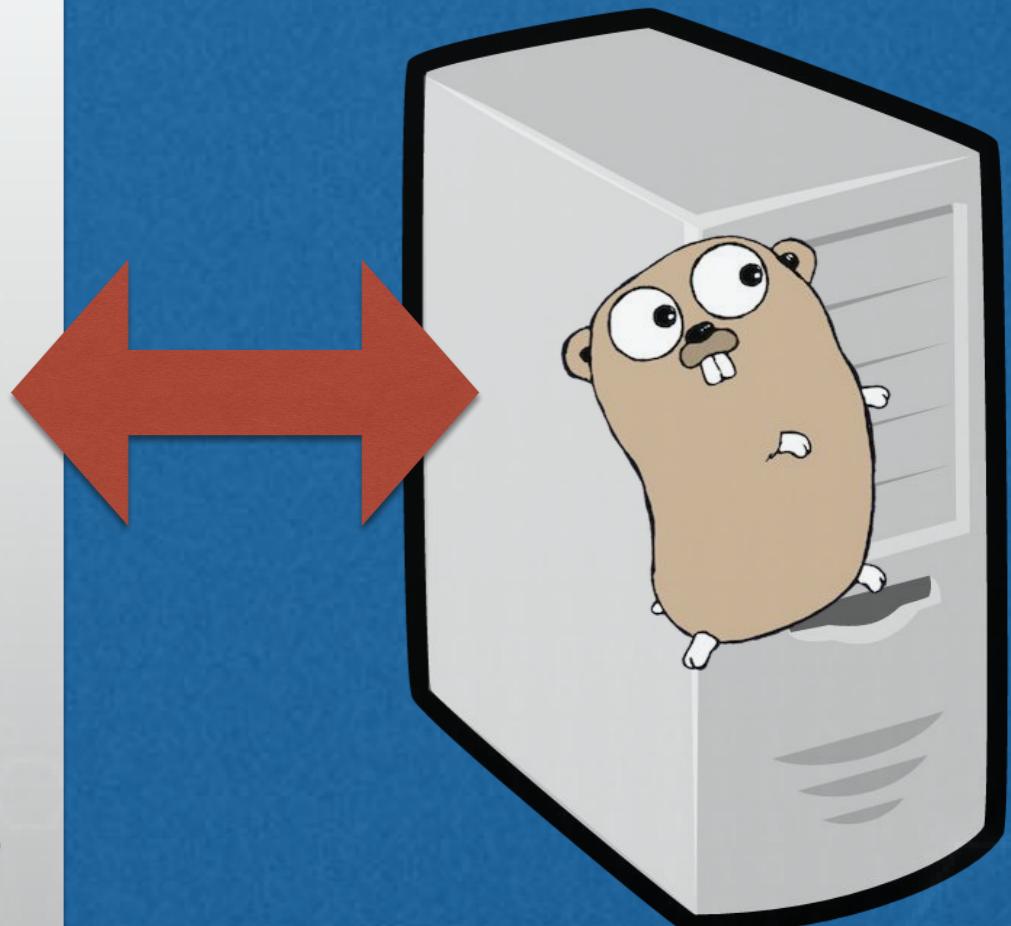
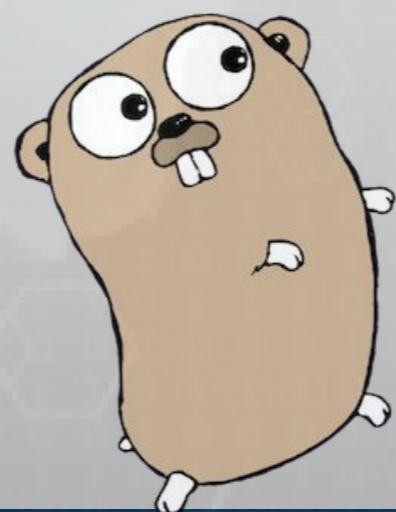
GoMobile

Using Golang in Mobile



Golang
Sample Console

**Golang
GoMobile
Package**





Golang IDE (Integrated Development Environment)

The screenshot shows a Visual Studio Code interface with the Go extension installed, demonstrating a debugging session for a Golang application named 'ptt'.

Code Editor: The main area displays the file `ptt.go`. A red dot at line 179 indicates a breakpoint has been set. The code implements a command-line application for interacting with a PTT board.

```
166     }
167     }
168 }
169 fmt.Printf("(o: open file in fider, s: top page, n:next, p:prev, quit: quit program)\n")
170 return hrefs
171 }
172
173 func main() {
174     var format = logging.MustStringFormatter(`{{level}} {{message}}`)
175     logging.SetFormatter(format)
176     logging.SetLevel(logging.INFO, "iloveptt")
177
178     usr, _ := user.Current()
179     baseDir = fmt.Sprintf("%v/Pictures/iloveptt", usr.HomeDir)
180
181     var workerNum int
182     rootCmd := &cobra.Command{
183         Use:   "iloveptt",
184         Short: "Download all the images in given post url",
185         Run: func(cmd *cobra.Command, args []string) {
186             page := 0
187             hrefs := parsePttBoardIndex(page)
188
189             scanner := bufio.NewScanner(os.Stdin)
190             quit := false
191
192             for !quit {
```

Variables View: The left sidebar shows the current variable state. A `usr` variable is listed under the `Local` section.

Watch View: The `WATCH` section shows the declaration of `workerNum` and the definition of `rootCmd`.

Call Stack: The `CALL STACK` view shows the stack trace, indicating the application is paused on a breakpoint at line 179 of `ptt.go`.

Debug Console: The bottom pane displays the output of the debugger, showing the process launching and breakpoints being created.

```
2016/04/18 00:49:42 debugger.go:61: launching process with args: ./debug
2016/04/18 00:49:42 debugger.go:228: created breakpoint: &api.Breakpoint{ID:1, Name:"", Addr:0x54b7, File:"/Users/Evan/src/go/src/github.com/kkdai/iloveptt/ptt.go", Line:221, FunctionName:"main.main.func1", Cond:"", Tracepoint:false, Goroutine:false, Stacktrace:0, Variables:[]string(nil), HitCount:map[string]uint64{}, TotalHitCount:0x0}
2016/04/18 00:49:42 debugger.go:228: created breakpoint: &api.Breakpoint{ID:2, Name:"", Addr:0x424f, File:"/Users/Evan/src/go/src/github.com/kkdai/iloveptt/ptt.go", Line:184, FunctionName:"main.main", Cond:"", Tracepoint:false, Goroutine:false, Stacktrace:0, Variables:[]string(nil), HitCount:map[string]uint64{}, TotalHitCount:0x0}
2016/04/18 00:49:42 debugger.go:228: created breakpoint: &api.Breakpoint{ID:3, Name:"", Addr:0x40a9, File:"/Users/Evan/src/go/src/github.com/kkdai/iloveptt/ptt.go", Line:179, FunctionName:"main.main", Cond:"", Tracepoint:false, Goroutine:false, Stacktrace:0, Variables:[]string(nil), HitCount:map[string]uint64{}, TotalHitCount:0x0}
2016/04/18 00:49:42 debugger.go:377: continuing
```

Bottom Status Bar: The status bar shows the current file is `ptt.go`, the line is 179, column is 1, tab size is 4, encoding is UTF-8, and there are no errors or warnings.

Visual Studio Code Go Extension



Build IM Line Bot



BOT API Trial Account

在開始使用BOT API之前，您可以先試用它的部分功能。

可試用的功能為BOT API的用戶交流功能。

Step 1: Request Line Bot Trial Account

Apply Line Bot trial account [here](#)
(Only 1 per Line Account)



Step 2: Deploy to Heroku by one click

LineBotTemplate: A simple Golang LineBot Template for Line Bot API

[chat on gitter](#)

[godoc reference](#) [build unknown](#)

Installation and Usage

1. Got A Line Bot API developer account

Make sure you already registered, if you need use Line Bot.

2. Just Deploy the same on Heroku

[Deploy to Heroku](#)

Remember your heroku, ID.

Link :

<https://github.com/kkdai/LineBotTemplate>



Step 2: Deploy to Heroku by one click

 LineBot
Line Bot API tester
kkdai/linebotTemplate#master

Remember this address

App Name (optional)
Leave blank and we'll choose one for you.

testbotforslide2016

Runtime Selection
Your app can run in your choice of region in the Common Runtime.

United States

Deploy for Free

Create app

Configure environment

Build app

Run scripts & scale dynos

Deploy to Heroku

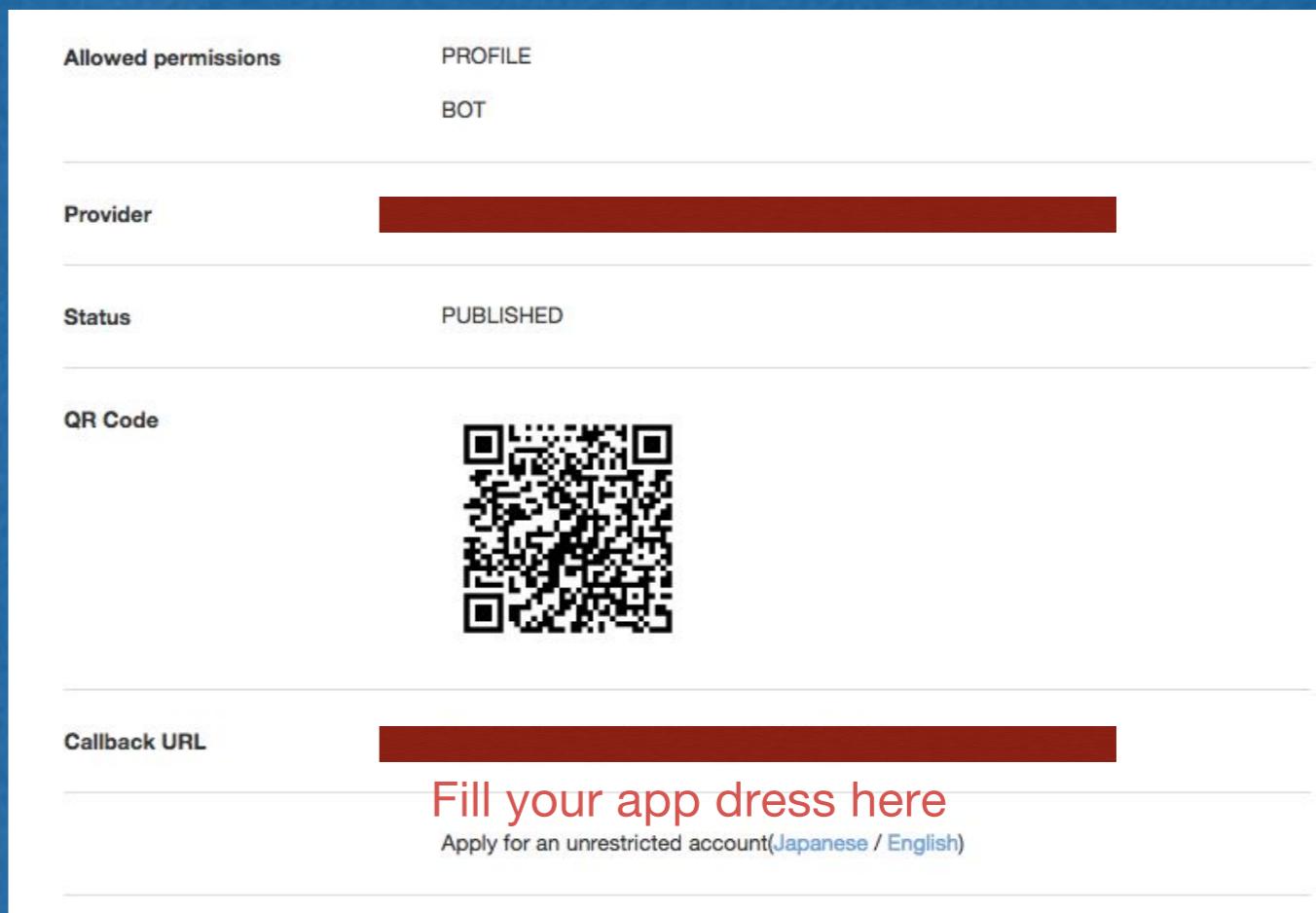
Your app was successfully deployed.

[Manage App](#) [View](#)



Step 3. Fill your Heroku App Callback address to Line Bot

- Copy your app address from Heroku to Line Dashboard.
- It will be something like "
<https://xxxxxx.herokuapp.com:443/callback>"





Step 4. Get related Line Bot info for Heroku App setting

- Need copy following variable:
 - Channel ID
 - Channel Secret
 - MID

台北流浪動物需要你
Need copy those three

Channel ID	[REDACTED]
Channel Secret	[REDACTED]
MID	[REDACTED]
Name	台北流浪動物需要你
Application type	TRIAL_BOT
App icon	



Step 5. Fill your Line Bot Information to Heroku Variables

- In Heroku [Dashboard] -> [Settings]-> [Config Variables]
- Create following variables
 - ChannelID
 - ChannelSecret
 - MID
- Fill with you just got from Line Bot Dashboard.

The screenshot shows the Heroku Dashboard with the app named 'petneedyou'. The 'Config Vars' section displays four environment variables:

Config Vars	Value
ChannelID	[REDACTED]
ChannelSecret	[REDACTED]
MID	uf73b5aae7048b7e0319916aee57ea697
KEY	VALUE



```
42 func callbackHandler(w http.ResponseWriter, r *http.Request) {
43     received, err := bot.ParseRequest(r)
44     if err != nil {
45         if err == linebot.ErrInvalidSignature {
46             w.WriteHeader(400)
47         } else {
48             w.WriteHeader(500)
49         }
50         return
51     }
52     for _, result := range received.Results {
53         content := result.Content()
54         if content != nil && content.IsMessage && content.ContentType == linebot.ContentTypeText {
55             bot.SendImage([]string{content.From}, "http://petneed.me/static/img/petNeedme_full_color.png", "htt
56         }
57     }
58 }
```

Parse bot request to received
result

Send a Image to user



```
42 func callbackHandler(w http.ResponseWriter, r *http.Request) {
43     received, err := bot.ParseRequest(r)
44     if err != nil {
45         if err == linebot.ErrInvalidSignature {
46             w.WriteHeader(400)
47         } else {
48             w.WriteHeader(500)
49         }
50     }
51     return
52 }
53 for _, result := range received.Results {
54     content := result.Content()
55     if content != nil && content.IsMessage && content.ContentType == linebot.ContentTypeText {
56         bot.SendImage([]string{content.From}, "http://petneed.me/static/img/petNeedme_full_color.png", "htt
57     }
58 }
```

Make sure it is text message not operation

Send a Image to user



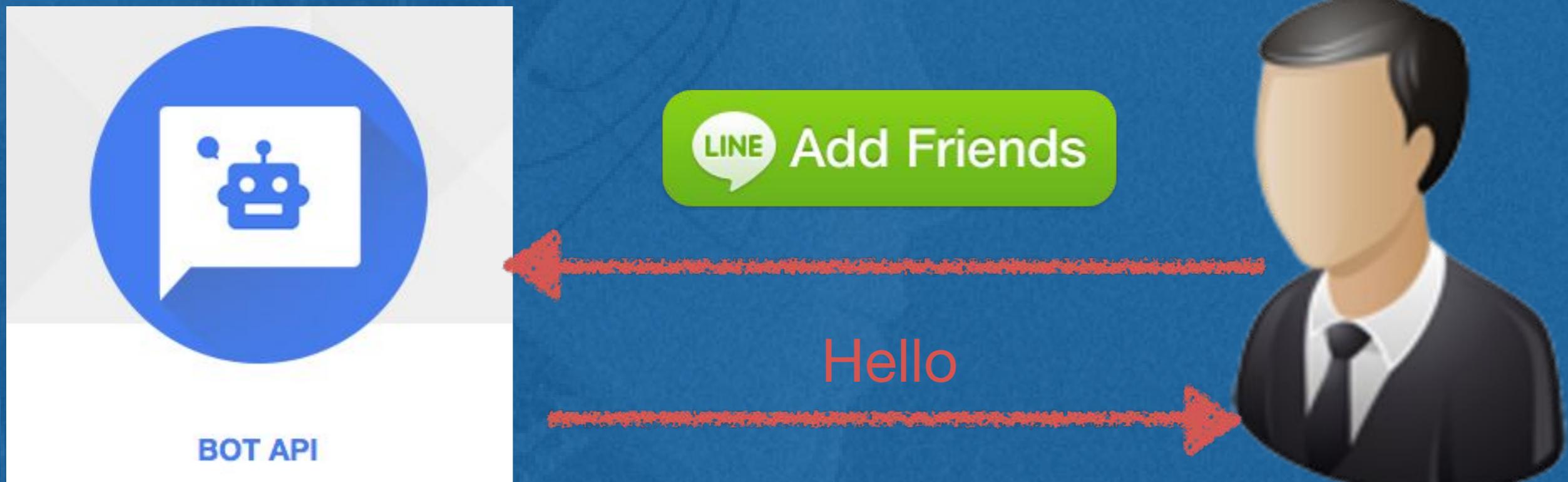
```
42 func callbackHandler(w http.ResponseWriter, r *http.Request) {
43     received, err := bot.ParseRequest(r)
44     if err != nil {
45         if err == linebot.ErrInvalidSignature {
46             w.WriteHeader(400)
47         } else {
48             w.WriteHeader(500)
49         }
50     }
51     return
52 }
53 for _, result := range received.Results {
54     content := result.Content()
55     if content != nil && content.IsMessage && content.ContentType == linebot.ContentTypeText {
56         bot.SendImage([]string{content.From}, "http://petneed.me/static/img/petNeedme_full_color.png", "htt
57     }
58 }
```

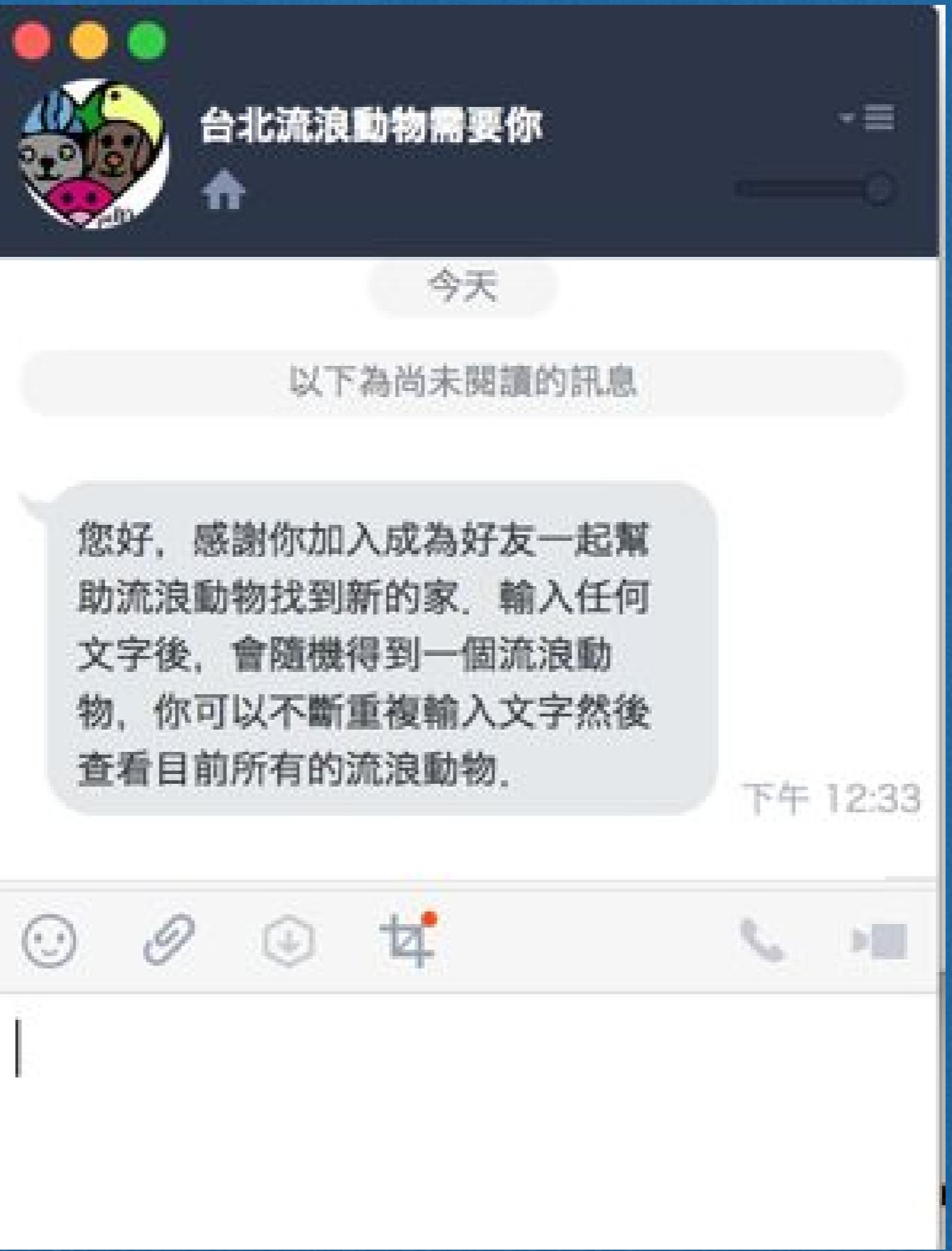
Send image to who text to the
bot

Send a Image to user



“Hello” message from Line Bot





```
47 func callbackHandler(w http.ResponseWriter, r *http.Request) {
48     received, err := bot.ParseRequest(r)
49     if err != nil {
50         if err == linebot.ErrInvalidSignature {
51             w.WriteHeader(400)
52         } else {
53             w.WriteHeader(500)
54         }
55         return
56     }
57     for _, result := range received.Results {
58         content := result.Content()
59
60         //Add with new friend.
61         if content != nil && content.IsOperation && content.OpType == linebot.OpTypeAddedAsFriend {
62             out := fmt.Sprintf("Hello Message!!!")
63             //result.RawContent.Params[0] is who send your bot friend added operation, otherwise you cannot get
64             _, err = bot.SendText([]string{result.RawContent.Params[0]}, out)
65             if err != nil {
66                 log.Println(err)
67             }
68             log.Println("New friend add, send cue to new friend.")
69         }
70     }
71 }
72 }
```

Determine the operation type

- Send a “Hello” message when add bot as friend

Expect Result

```
1 {"result":[]  
2  
3 {  
4     "from":"u2ddf2eb3c959e561f6c9fa2ea732e7eb8",  
5     "fromChannel":1341301815,  
6     "to":["u0cc15697597f61dd8b01cea8b027050e"],  
7     "toChannel":1441301333,  
8     "eventType":"138311609000106303",  
9     "id":"ABCDEF-12345678901",  
10    "content": {  
11        "location":null,  
12        "id":"325708",  
13        "contentType":1,  
14        "from":"uff2aec188e58752ee1fb0f9507c6529a",  
15        "createdTime":1332394961610,  
16        "to":["u0a556cffd4da0dd89c94fb36e36e1cdc"],  
17        "toType":1,  
18        "contentMetadata":null,  
19        "text":"Hello, BOT API Server!"  
20    }  
21 },  
22 ...  
23 }]
```

Expect Result

```
1 {
2     "result": [
3         {
4             "content": {
5                 "params": [
6                     "████████████████████████████████████",
7                     null,
8                     null
9                 ],
10                "message": null,
11                "reqSeq": 0,
12                "revision": 2196,
13                "opType": 8
14            },
15            "createdTime": 1473651726342,
16            "eventType": "138311609100106403",
17            "from": "████████████████████████████",
18            "fromChannel": 1341301815,
19            "id": "WB1521-3805928475",
20            "to": [
21                "████████████████████████████"
22            ],
23            "toChannel": 1462189304
24        }
25    ]
26 }
```

```
47 func callbackHandler(w http.ResponseWriter, r *http.Request) {
48     received, err := bot.ParseRequest(r)
49     if err != nil {
50         if err == linebot.ErrInvalidSignature {
51             w.WriteHeader(400)
52         } else {
53             w.WriteHeader(500)
54         }
55         return
56     }
57     for _, result := range received.Results {
58         content := result.Content()
59
60         //Add with new friend.
61         if content != nil && content.IsOperation && content.OpType == linebot.OpTypeAddedAsFriend {
62             out := fmt.Sprintf("Hello Message!!")
63             //result.RawContent.Params[0] is who send your bot friend added operation, otherwise you cannot get
64             _, err = bot.SendText([]string{result.RawContent.Params[0]}, out)
65             if err != nil {
66                 log.Println(err)
67             }
68             log.Println("New friend add, send cue to new friend.")
69         }
70     }
71 }
72 }
```

Get request from ID, this is only way for now

Send a “Hello” message when add bot as friend

Example: PetNeedMe

<https://github.com/kkdai/LineBotPetNeedM>





Build Facebook Bot



Step 1: Deploy to Heroku

FBBotTemplate: A simple Golang Facebook Bot Template

[chat on gitter](#)

[godoc](#) [reference](#) [build](#) [unknown](#)

[go report](#) [B](#)

Installation and Usage

1. Just Deploy the same on Heroku

 [Deploy to Heroku](#)

Remember your heroku ID and app address. ex: https://APP_ADDRESS.herokuapp.com/

Link :

<https://github.com/kkdai/FBBotTemplate>



Deploy your own

FBBot

FB Bot Template

[kkdai/FBBotTemplate#master](#)

Remember this address

App Name (optional)

Leave blank and we'll choose one for you.

fbbotformodernweb

Runtime Selection

Your app can run in your choice of region in the
Common Runtime.

United States

Deploy for Free

Create app



Configure environment



Build app



Run scripts & scale dynos



Deploy to Heroku





Step 2: Create Facebook Page

The screenshot shows a Facebook mobile interface. At the top, there is a blue header bar with a profile picture of a man, the name "Evan Lin", and a "首頁" button. To the right of the header are icons for people, messages, notifications (with a red box containing the number 1), a lock, and a dropdown menu. Below the header, a white sidebar on the left lists "你的粉絲專頁" (Your Fan Pages) and "Petnee". A large central box displays "你的專頁：" (Your Page) with two entries: "Petneedme" and "Linker Networks". The "Petneedme" entry includes a small profile picture of three dogs and a "1" in a box. The "Linker Networks" entry includes a briefcase icon and a "1 則訊息" (1 message) badge. At the bottom of this central box are two blue buttons: "建立粉絲專頁" (Create Fan Page) and "管理粉絲專頁" (Manage Fan Page). The background of the entire screen is a dark blue gradient.

你的專頁：

Petneedme

Linker Networks 1 則訊息

建立粉絲專頁

管理粉絲專頁



Step 3: Create App for Messenger

- Create new app
“Web App”
- Select new
product
“Messenger”





Step 4: Create App Token

- Select generate token.
- Need a “Page”

權杖產生

必須提供粉絲專頁權杖才能開始使用 API。此粉絲專頁權杖將擁有所有 Messenger 權限，即使你的應用程式尚未獲准使用這些權限（但是在這個情況下，你只能傳送訊息給應用程式管理員）。你也可以使用 Facebook 登入，為並非你擁有的粉絲專頁產生粉絲專頁權杖。

粉絲專頁

選擇粉絲專頁 ▾

粉絲專頁存取權杖

You must select a Page to generate an access token.



Step 5: Paste token to Heroku Variable

Overview Resources Deploy Metrics Activity Access Settings

Name

a11111assssss



Config Variables

Config vars change the way your app behaves. In addition to creating your own, some add-ons come with their own.

Config Vars

There are no config vars for this app yet

[Learn about config vars](#) in the Dev Center.

TOKEN

VALUE

Add



Step 6: Setup Page Subscription for Messenger Bot

新 Page 訂閱

x

回呼網址

https://APP_ADDRESS.herokuapp.com/webhook

https://www.example.com/

k

驗證權杖

Token we just
generate

訂閱欄位

message_deliveries

messages

messaging_optins

messaging_postbacks

messaging_account_linking

取消

驗證並儲存



Step 7: Select a Page for your Webhooks

Webhooks

Edit events

若要接收 Messenger 用戶傳送的訊息和其他活動，應用程式必須啟用 Webhooks 整合。



選擇的活動： message_deliveries, messages

Select a page to subscribe your webhook to the page events

選擇粉絲專頁 ▾

已訂閱的粉絲專頁： Linker Networks



Step 8: Select a Page for your Webhooks

Messenger 應用程式審查

[Request Permissions](#)

若要使用 Messenger 平台，你的應用程式必須獲准使用傳送／接收 API (pages_messaging)。傳送／接收 API 只能用來傳送自然生成的內容，不應用來傳送行銷或其他推廣性質的訊息。若要使用現有聯絡人的電話號碼尋找 Messenger 帳號，你需要使用「客戶比對」(pages_messaging_phone_number)。

pages_messaging

Enables your app to send and receive messages using a Facebook Page.
This permission cannot be used to send promotional or advertising content.

未批准

pages_messaging_phone_number

Enables your app to match phone numbers you already have to Messenger accounts. In order to use this permission, your app must be already approved for the Send/Receive API and based in the US.

未批准



Example: PetNeedMe

我們不想再流浪
期待有個溫暖的家

Petneedme
社區

首頁 關於 按讚分析 相片

Petneedme
Evan Lin 你好，這裡有寵物等著你領養: 您好，目前的動物：
名為, 所在地為:臺北市動物保護處105062415, 敘述: 電話
為:02-87913063

Petneedme
2016-7-08 0:33

2016-7-08 0:33

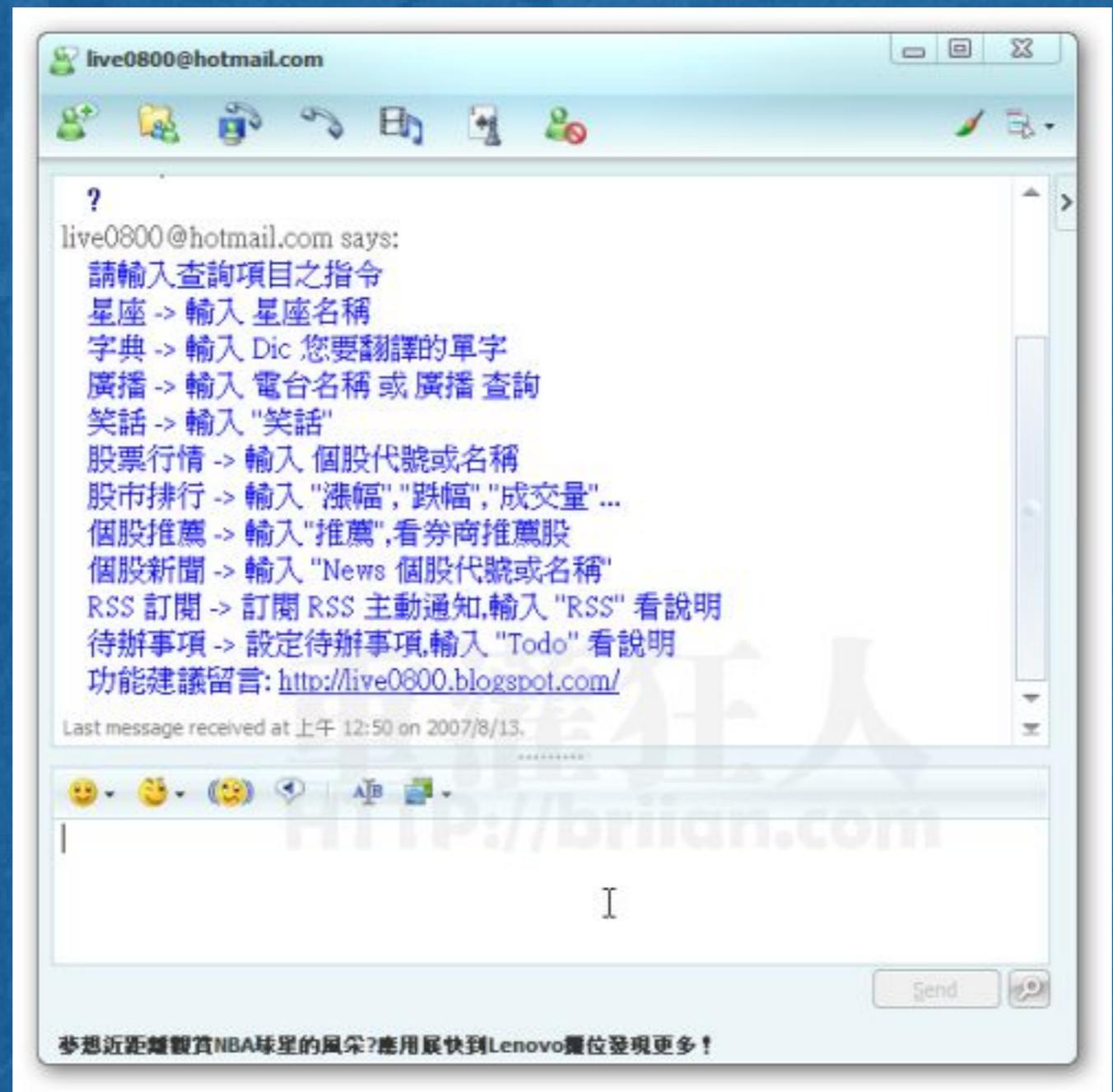


你的 IM 機器人夠聰明嗎？



IM Bot 10 years ago

“有問必達”



圖片重灌狂人:

<https://brilian.com/2536/msn-bot.html>



IM Bot 10 years later “Fugle股市小幫手”

● Fugle股市小幫手 ⚙️ ×

麼回答，我會盡快學習相關的知識！
您可以輸入【Help】來查看我可以為您提供什麼服務！

 [直接看說明](#)

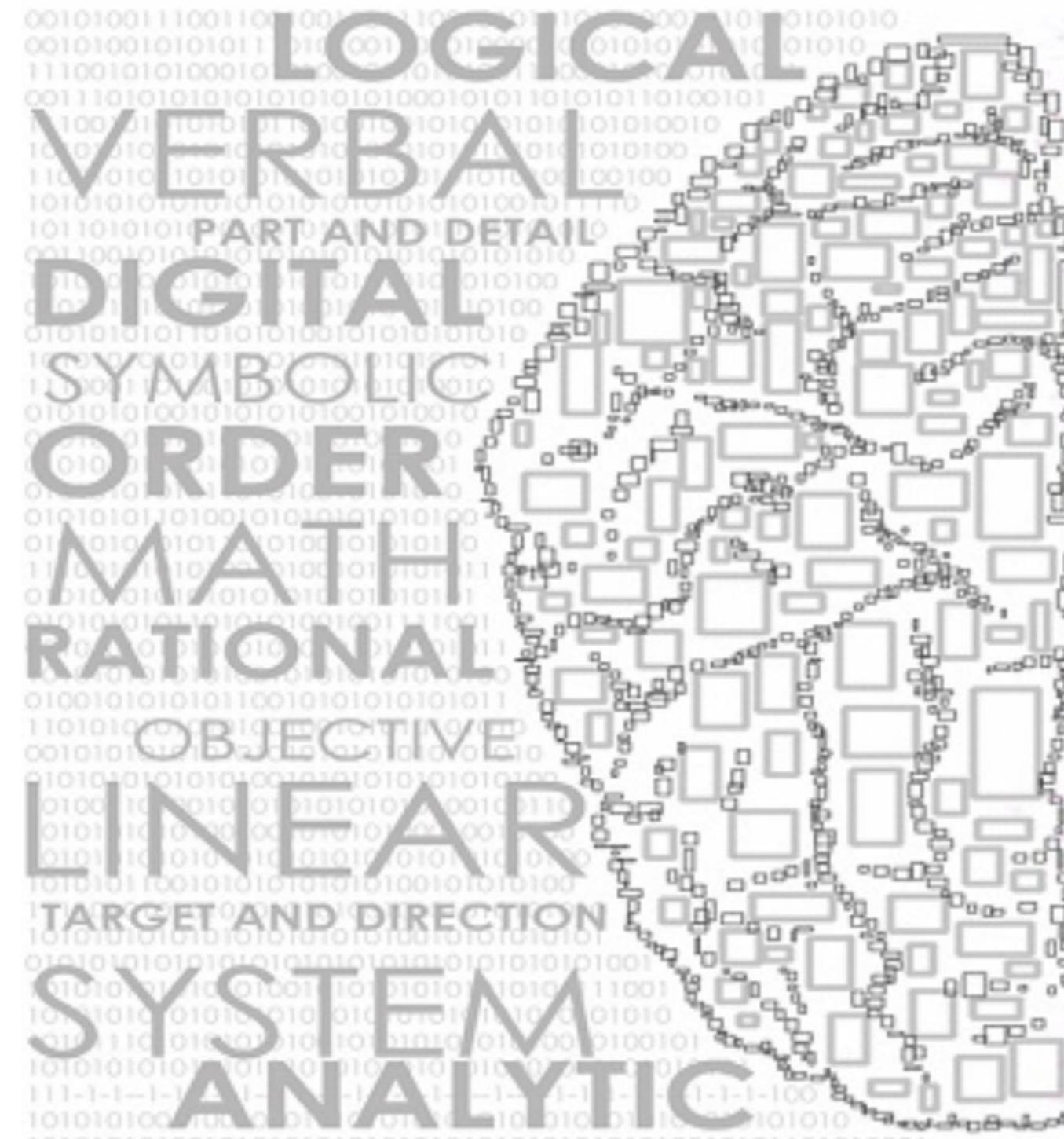
 [請問您想瞭解什麼功能呢？](#)

[自選追蹤](#) [股票資訊](#) [看全部說明](#)

≡



LOGICAL
VERBAL
PART AND DETAIL
DIGITAL
SYMBOLIC
ORDER
MATH
RATIONAL
OBJECTIVE
LINEAR
TARGET AND DIRECTION
SYSTEM
ANALYTIC



- Left Brain
- Logical
- Verbal
- Language



台北天氣如何？



台北天氣如何？

台北有下雨嗎？



台北天氣怎麼樣?

去台北需要帶傘嗎?

台北天氣如何?

台北有下雨嗎?

台北天空是什麼顏色?

台北天氣好嗎?

台北冷嗎?



台北天氣如何？

Intent: ??

Entity: ??



台北天氣如何？

Intent: Check Weather

Entity: ??



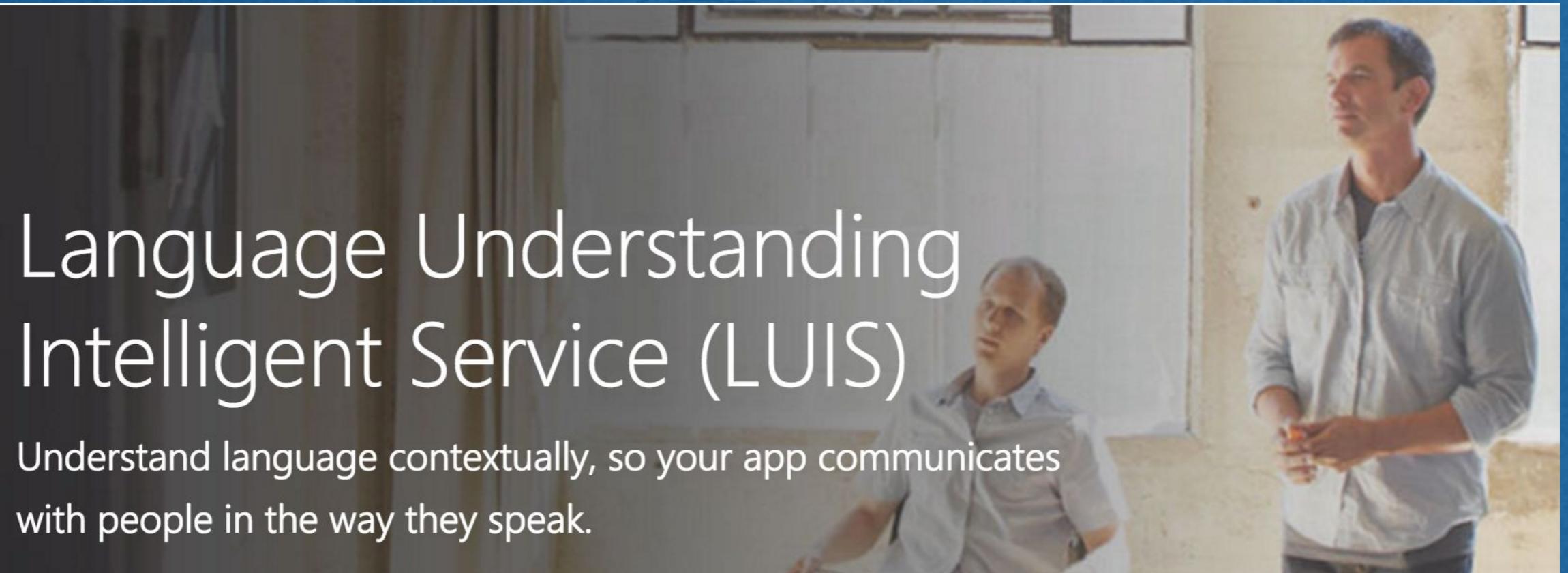
台北天氣如何？

Intent: Check Weather

Entity: Taipei



Use LUIS to build smart Bot



Language Understanding Intelligent Service (LUIS)

Understand language contextually, so your app communicates
with people in the way they speak.





Use LUIS to build smart Bot

LUIS

Go to Preview My Applications About Help Docs Support Evan Lin Sign Out

Bot

New utterances Search Suggest Review labels Performance analysis

App Settings

Publish

Intents

None
FindNews
AskWeather

Entities

Topic
天氣
Date

Pre-built Entities

No pre-built entities added

Regex Features

No patterns added

Phrase List Features

No phrase list features added

Train

Find some news about Evan

Find some news about Evan

FindNews(0.55)

Submit

Intents

FindNews
14 utterances: 14 correctly predicted

None
10 utterances: 10 correctly predicted

AskWeather
2 utterances: 0 correctly predicted

Correctly predicted
Error (predicted as other intent)

Microsoft



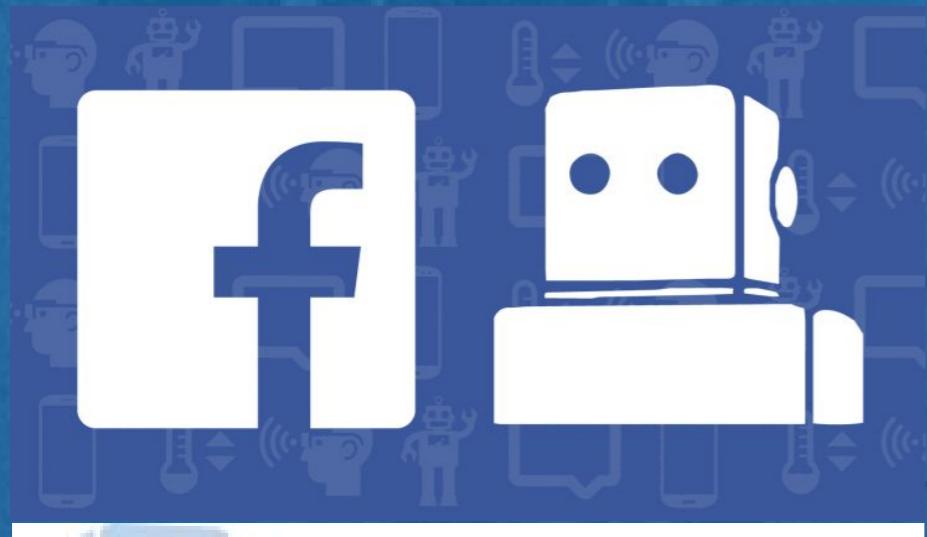
LUIS

```
{  
  "query": "Find some news about Evan",  
  "intents": [  
    {  
      "intent": "FindNews",  
      "score": 0.5175246  
    },  
    {  
      "intent": "None",  
      "score": 0.2611109  
    },  
    {  
      "intent": "AskWeather",  
      "score": 2.99510958E-08  
    }  
],  
  "entities": [  
    {  
      "entity": "Evan",  
      "type": "Topic",  

```



Other Natural Language Processing Tool



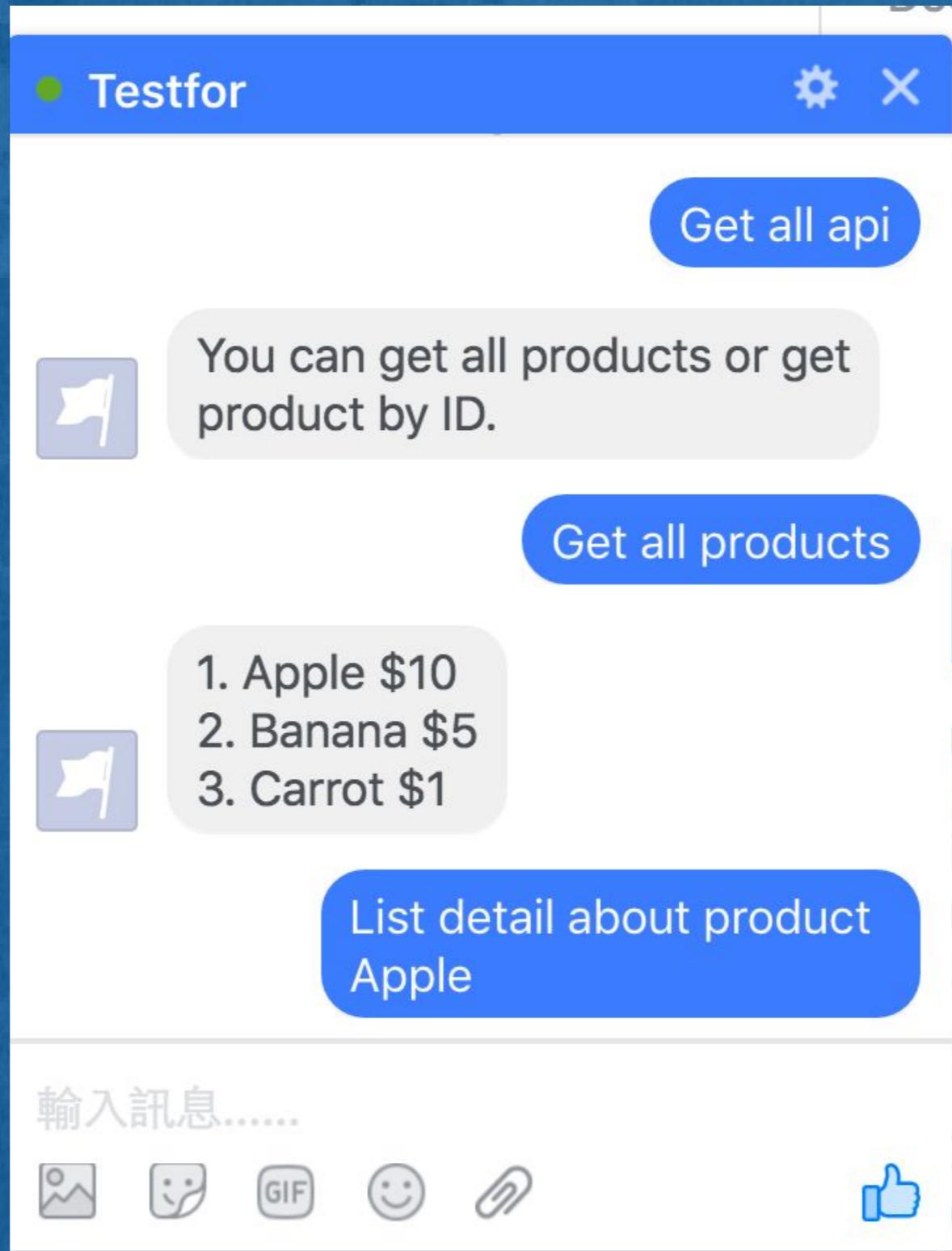


Future





Future : Bot to Bot (No more API)





Future : Customer Service Bot



柯維然 ▶ g0v.tw 後勤中心
9月11日 ·

大家好，我目前服務於新竹市警察局
常聽到很多民眾詢問
我要怎麼申請？派出所在哪裡？罰單怎麼繳？
有時候連自己要找個申請流程
都要在網站找個半天
是不是能直接用Line問呢！

所以就用Line API 開發了一個給民眾的Q&A機器人
不過還沒正式上線
code整理好後會丟出來
也給其他政府機關參考



Future : For Your Website and Business



Evan Lin

2:45

什麼是 DC/OS ?



Testfor

2:46

DC/OS 是一個 DC/OS 是第一個 100% 開源的企業級 Datacenter Operating System · 歡迎到這裡查看更多 的資訊 <http://dcos.tw/>

DC/OS Taiwan

DC/OS 是第一個 100% 開源的企業級 Datacenter Operating Sys...
dcos.tw



Evan Lin

2:46

有沒有 DC/OS 的教學手冊?



Testfor

2:48

DC/OS 安裝很簡單，不論從 Vagrant 或是 GCE, AWS 都可以 · 歡迎查案以下資料:

1. Install from Vagrant

<https://dcos.io/docs/1.7/administration/installing/local-mode-deployment/>



Future : For Your Website and Business



Evan Lin

2:48

我們公司想導入 DC/OS 有人能幫助我們嗎?



Testfor

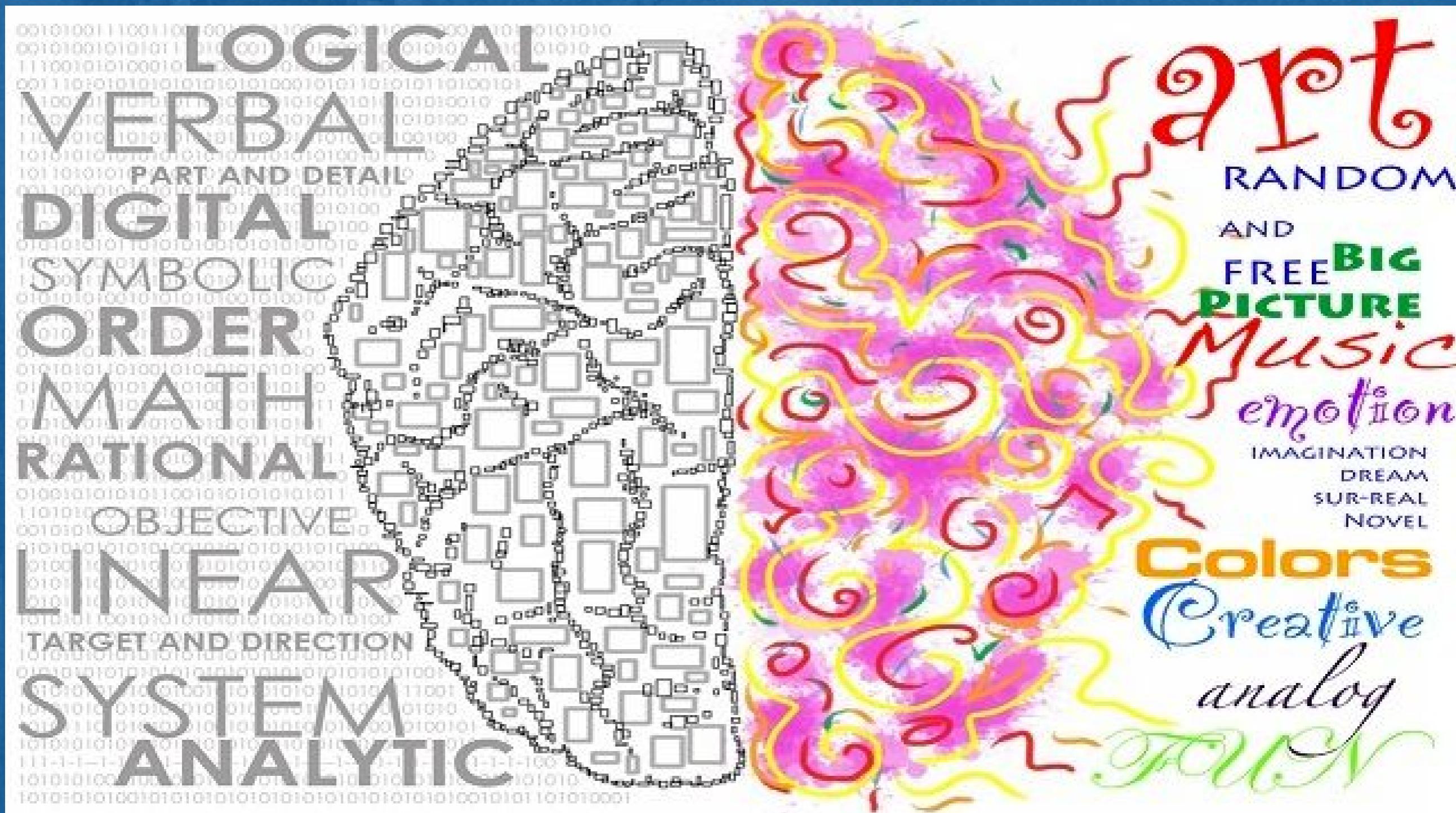
2:49

您好，我們已經將你的需求轉告給 Linker 相關人員。
會有人盡快與你聯絡...





Future : How About Build Right Brain?



We are Hiring..

1. Senior Data Scientist_{Fr}
2. Senior Front-end

Unleashing Machine Intelligence

In Data Center / IoT / Mobile Networks

[Contact US](#)



歡迎訂閱 碼天狗



CodeTengu

LATEST ISSUE

ARCHIVES

SPONSORSHIP



CodeTengu Weekly（碼天狗週刊）是一份開發者導向的 IT 技術週刊，適合所有患有資訊焦慮症、氣血循環不順以及性受挫的軟體工程師們。

致力於解決開發者之間的資訊不對稱。

<http://weekly.codetengu.com/>