# 用 Kamigo 實作 LINE bot 的開發心得

NiJia

2019/11/27

Chatbot Taiwan #14 meetup
@ onramp studio

# 林家煜 NiJia

- Hyena - Backend Developer

- Chatbot Taiwan - Co-organizer

- LINE API Expert

- Farmer / Body builder

- https://nijialin.com

- louis70109@gmail.com

這是我做的肌肉仔

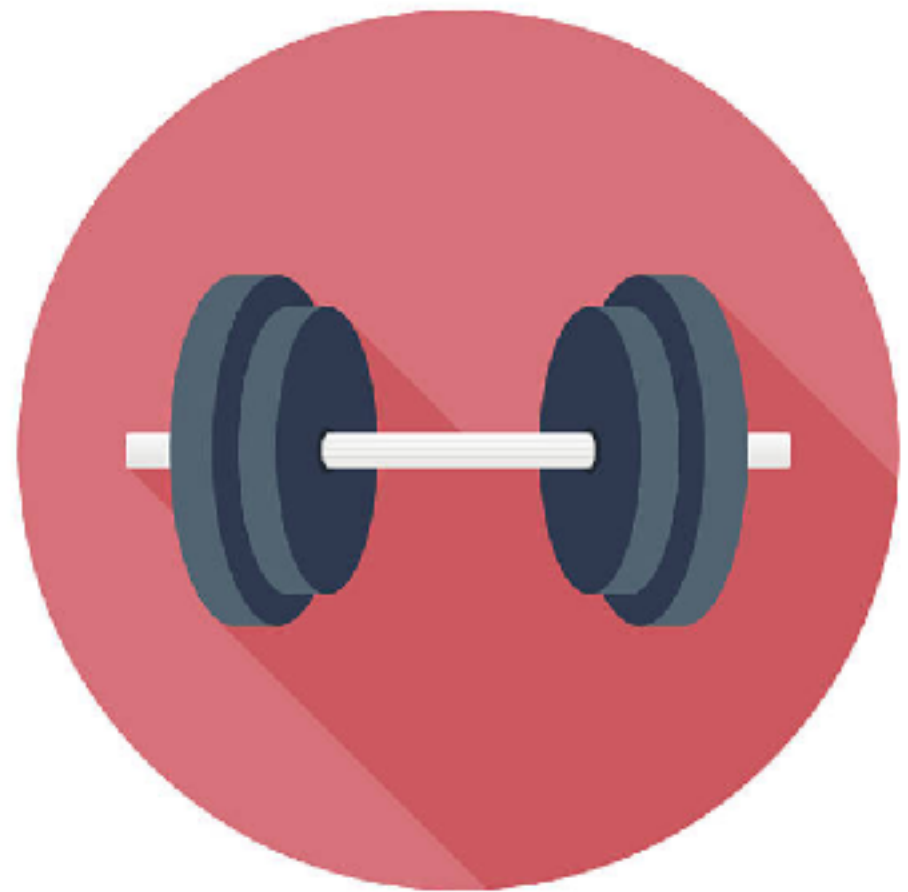https://github.com/louis70109/muscle_man

# 為什麼要做成機器人

- 本身是重度 LINE 使用者

- 熊大、莎莉愛好者

- 不太愛安裝一堆 APP (懶)

- 做成網頁誰要用 **(通常都是有教練在幫忙用)**

# 肌肉仔期望功能

- 要能 CRUD

- 隨身一隻手機

- 練完就紀錄

- 需要知道訓練時間

- 紀錄的身份驗證

- 區間紀錄搜尋

- 上次訓練紀錄

# 肌肉仔功能

- 要能 CRUD　　<— 基本的

- 隨身一隻手機

- 練完就紀錄

- 需要知道訓練時間

- 紀錄的身份驗證

- 區間紀錄搜尋

- 上次訓練

依照身體狀態計畫課表(X)
讓訓練課表變得可見
像我時間偶爾練會需要有東西幫我記錄
上次訓練

# 肌肉仔功能

- 要能 CRUD

- **隨身一隻手機** <— 紀錄用 (很多人都佔著器材追劇)

- 練完就紀錄

- 需要知道訓練時間

- 紀錄的身份驗證

- 區間紀錄搜尋

- 上次訓練

# 肌肉仔功能

- 要能 CRUD

- 隨身一隻手機

- **練完就紀錄** **<- 救救金魚腦**

- 需要知道訓練時間

- 紀錄的身份驗證

- 區間紀錄搜尋

- 上次訓練

依照身體狀態計畫課表(X)
讓訓練課表變得可見
像我時間偶爾練會需要有東西幫我記錄
上次訓練

# 肌肉仔功能

- 要能 CRUD

- 隨身一隻手機

- 練完就紀錄

- 需要知道訓練時間   <— 控制狀態

- 紀錄的身份驗證

- 區間紀錄搜尋

- 上次訓練

# 肌肉仔功能

- 要能 CRUD

- 隨身一隻手機

- 練完就紀錄

- 需要知道訓練時間

- 紀錄的身份驗證　<ー之後做分析才有根據

- 區間紀錄搜尋

- 上次訓練

# 肌肉仔功能

- 要能 CRUD

- 隨身一隻手機

- 練完就紀錄

- 需要知道訓練時間

- 紀錄的身份驗證

- **區間紀錄搜尋** <ー特定紀錄 & 某次紀錄

- 上次訓練

# 肌肉仔功能

- 要能 CRUD

- 隨身一隻手機

- 練完就紀錄

- 需要知道訓練時間

- 紀錄的身份驗證

- 區間紀錄搜尋

- 上次訓練　<ー救救金魚腦

快速展示 肌肉仔

今天

已讀
下午 12:10    首頁

| 首頁 | 過程 | 詳細 | 新增 |
|------|------|------|------|

找不到訓練內容，快去練！

上次訓練

下午 12:10

單項
成績

新增

首頁

查詢
日期

上次
訓練

紀錄是我的本命

新增訓練項目
muscle-man.herokuapp.com

✕

類型

橫鈴 ▾

種類

請選擇 ▾

動作

▾

次數

1 ▾

組數

1 ▾

∧ ∨ 　　　　　　　　　　完成

胸
背
**腿**
肩膀
手臂
核心

# 新增訓練項目
muscle-man.herokuapp.com

×

類型

槓鈴 ▼

種類

腿 ▼

動作

深蹲 ▼

次數

1 ▼

組數

1 ▼

∧ ∨                    完成

羅馬尼亞
腿推
腿彎舉
**股二頭抬腿**
腿內收
腿外展
小腿腿
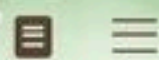
總共訓練時間約：0時0分30秒

下午 12:11

已讀
下午 12:11　首頁

| 首頁 | 過程 | 詳細 | 新增 |
|------|------|------|------|
| 訓練項目 | 重量 (kg) | | |
| 槓鈴股二… | 123.0 | | ✖ |
| 槓鈴下胸 | 43.0 | | ✖ |

總共訓練時間約：0時0分43秒

下午 12:11

已讀
下午 12:11　首頁

| 首頁 | 過程 | 詳細 | 新增 |
|------|------|------|------|
| 訓練項目 | 重量 (kg) | | |
| 槓鈴股二… | 123.0 | | ✖ |
| 槓鈴下胸 | 43.0 | | ✖ |
| 槓鈴坐式… | 21.0 | | ✖ |

總共訓練時間約：0時0分56秒

下午 12:11

⌨ 紀錄是我的本命

‹ 34 🛡 肌肉仔
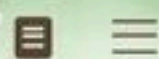
| 訓練項目 | 重量 (kg) | |
|---|---|---|
| 槓鈴股二… | 123.0 | ✗ |
| 槓鈴下胸 | 43.0 | ✗ |
| 槓鈴坐式… | 21.0 | ✗ |

總共訓練時間約：0 時 0 分 56 秒

下午 12:11

已讀
下午 12:11　今日訓練

🚀 槓鈴股二…　　　　刪除

| 類型 | 槓鈴 |
|---|---|
| 動作 | 股二頭抬腿 |
| 重量 (kg) | 123.0 |
| 次數 | 1 |
| 組數 | 1 |
| 組間 (分) | 0.5 |
| 建立時間 | 2019-11-09 |

回首頁　　　　過程

🚀 槓鈴

類型
動作
重量 (k
次數
組數
組間 (分
建立時間

回首頁

紀錄是我的本命

🚀 槓鈴股二…　　　　　刪除

| 類型 | 槓鈴 |
|---|---|
| 動作 | 股二頭抬腿 |
| 重量 (kg) | 123.0 |
| 次數 | 1 |
| 組數 | 1 |
| 組間 (分) | 0.5 |
| 建立時間 | 2019-11-09 |

回首頁　　　　過程

下午 12:11

已讀　過程
下午 12:12

總共訓練時間約： 0時 0分 56秒

12:11　　○　🚀 槓鈴股二頭抬腿

12:11　　○　🚀 槓鈴下胸

12:11　　○　🚀 槓鈴坐式划船

回首頁　　　　今日訓練

紀錄是我的本命

POST /practices/search

查詢項目
muscle-man.herokuapp.com
✕

## 請選擇日期

2019 ▼

November ▼

9 ▼

送出

< 18 🛡 肌肉仔

下午1:23

POST /practices/search {"trip": {"departing(1i)":"2019","departing(2i)":"10","departing(3i)":"14"}}

已讀
下午1:24

| 首頁 | | 2019-10-14 |
|---|---|---|
| **訓練項目** | **重量 (kg)** | |
| 史密斯平胸 | 50.0 | ✗ |
| 史密斯上胸 | 40.0 | ✗ |
| 啞鈴夾胸 | 16.0 | ✗ |
| Cable 三… | 18.7 | ✗ |
| Cable 三頭 | 6.8 | ✗ |
| Cable 平胸 | 4.6 | ✗ |

總共訓練時間約： 0 時 40 分 50 秒

單項成績　新增　首頁　查詢日期　上次訓練

紀錄是我的本命

首頁　　　　　　　　歷史成績

## 查詢歷史項目
muscle-man.herokuapp.com　　✕

查詢前 N 個月

5 ▼

項目

啞鈴二頭彎舉 ▼

送出

∧ ∨　　　　　　　　　　　完成

啞鈴二頭彎舉
**啞鈴平胸**
機械式上腹肌
機械式夾胸
臥姿斯平胸

| Cable 三頭 | 6.8 | ✕ |
|---|---|---|
| Cable 平胸 | 4.6 | ✕ |

總共訓練時間約：0 時 40 分 50 秒

下午 12:21

POST /practices/history
{"month":"5","traning":"1"
}

已讀
下午 12:22

| 首頁 | | 歷史成績 |
|---|---|---|
| 項目 | 重量 (kg) | 日期 |
| 啞鈴平胸 | 56.0 | 19-08-08 |
| 啞鈴平胸 | 52.0 | 19-08-26 |
| 啞鈴平胸 | 40.0 | 19-09-02 |
| 啞鈴平胸 | 60.0 | 19-09-09 |
| 啞鈴平胸 | 40.0 | 19-10-09 |

單項成績　　新增　　首頁　　查詢日期　　上次訓練

紀錄是我的本命

‹ 19 🛡 肌肉仔 📋 ☰
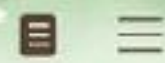
已讀
下午1:24　上一次練了什麼？

| 首頁 | 2019-10-14 |
|---|---|
| 訓練項目 | 重量 (kg) | |
| 史密斯平胸 | 50.0 | ✗ |
| 史密斯上胸 | 40.0 | ✗ |
| 啞鈴夾胸 | 16.0 | ✗ |
| Cable 三… | 18.7 | ✗ |
| Cable 三頭 | 6.8 | ✗ |
| Cable 平胸 | 4.6 | ✗ |

下午1:24

已讀
下午1:24　GET /practices/247

| 首頁 | 修改 | 刪除 |
|---|---|---|
| 類型 | | 史密斯 |
| 動作 | | 上胸 |
| 重量 (kg) | | 40.0 |
| 次數 | | 10 |
| 組數 | | 3 |
| 組間 (分) | | 1.5 |
| 建立時間 | | 2019-10-14 |

下午1:24

⌨ 　　　紀錄是我的本命

Kamigo 是怎麼誕生的？

Do Re Mi So

<> Code    ⊘ Issues 0    ⑂ Pull requests 0    ▥ Projects 0    📖 Wiki    🛡 Security    📊 Insights

a chatbot framework based on rails

| ⓣ 25 commits | ⑂ 1 branch | ◇ 0 releases | 👥 1 contributor | ⚖ MIT |
|---|---|---|---|---|

Branch: master ▾    New pull request                    Create new file   Upload files   Find file   Clone or download ▾
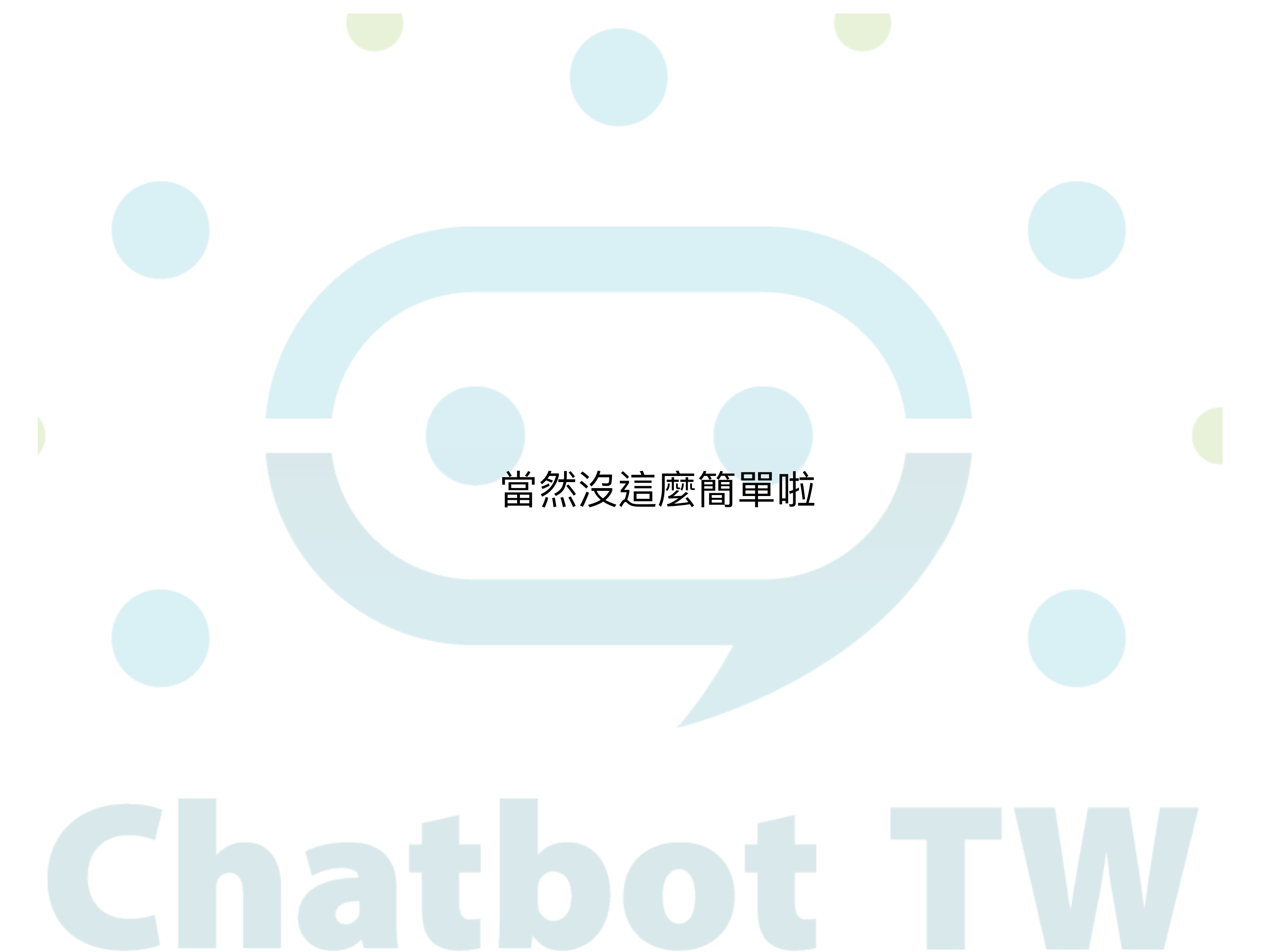
| 茶 **etrex** update readme | | Latest commit 2498e96 3 days ago |
|---|---|---|
| 📁 app | fix: can't get platform params from kamigo | last month |
| 📁 bin | git init | 4 months ago |
| 📁 config | first version | 4 months ago |
| 📁 doc | update readme | 3 days ago |
| 📁 lib | fix: can't get platform params from kamigo | last month |
| 📁 test | git init | 4 months ago |
| 📄 .gitignore | extract event parser | 2 months ago |
| 📄 Gemfile | git init | 4 months ago |
| 📄 Gemfile.lock | update gemfile | 15 days ago |
| 📄 MIT-LICENSE | git init | 4 months ago |
| 📄 README.md | update readme | 3 days ago |

當然沒這麼簡單啦

作者在台下

等等可以問他🤣

還不來按個星星？ **https://github.com/etrex/kamigo**

- Rails 的黑魔法 - Scaffold: CRUD 快速集成

- MVC 架構

  - 一個請求對應一個 controller

  - LIFF & Flex message 都在 View 裡

- 目前 Rails 6.0.x 會爆

- Rails 的黑魔法 - Scaffold: CRUD 快速集成

```
[muscle_man] rails g scaffold todo name desc
```

MVC 架構

- 一個請求對應一個 controller

- LIFF & Flex message 都在 View 裡

- 目前 Rails 6.0.x 會爆

- Rails 的黑魔法 - Scaffold: CRUD 快速集成

- MVC 架構

  - 一個請求對應一個 controller

  - LIFF & Flex message 都在 View 裡

- 目前 Rails 6.0.x 會爆

- Rails 的黑魔法 - Scaffold: CRUD 快速集成

- MVC 架構

  - 一個請

- LIFF &

- 目前 Ra

這個 ➡️ 已讀 首頁
下午 3:51

| 首頁 | 過程 | 詳細 | 新增 |

找不到訓練內容，快去練！

上次訓練

下午 3:52

- Rails 的黑魔法 - Scaffol
- MVC 架構
  - 一個請求對應一個 co
  - LIFF & Flex message
- 目前 Rails 6.0.x 會爆

- Rails 的黑魔法 - Scaffold: CRUD 快速集成

- MVC 架構

```
respond_to do |format|
  format.html { redirect_to @practices, notice: 'Practice was successfully
  format.json { render :show, status: :created, location: @practices }
  format.line { render :search }
end
```

- LIFF & Flex message 都在 View 裡

- 目前 Rails 6.0.x 會爆

- Rails 的黑魔法 - Scaffold: CRUD 快速集成

- MVC 架構

  - 一個請求對應一個 controller

  - LIFF & Flex message 都在 View 裡

- 目前 Rails 6.0.x 會爆 (某次 Live demo ...)

# Kamigo 宗旨

可以同時開發 Chatbot 與 Web

並共用 Controller 以及 Model

設計理念: https://github.com/etrex/kamigo/blob/master/doc/01_intro.md

# Kamigo 宗旨

可以同時開發 Chatbot 與 Web

並共用 Controller 以及 Model

但我只用來開發 Chatbot !!

設計理念: https://github.com/etrex/kamigo/blob/master/doc/01_intro.md

# Kamigo 宗旨

~~可以同時開發 Chatbot 與 Web~~

~~並共用 Controller 以及 Model~~
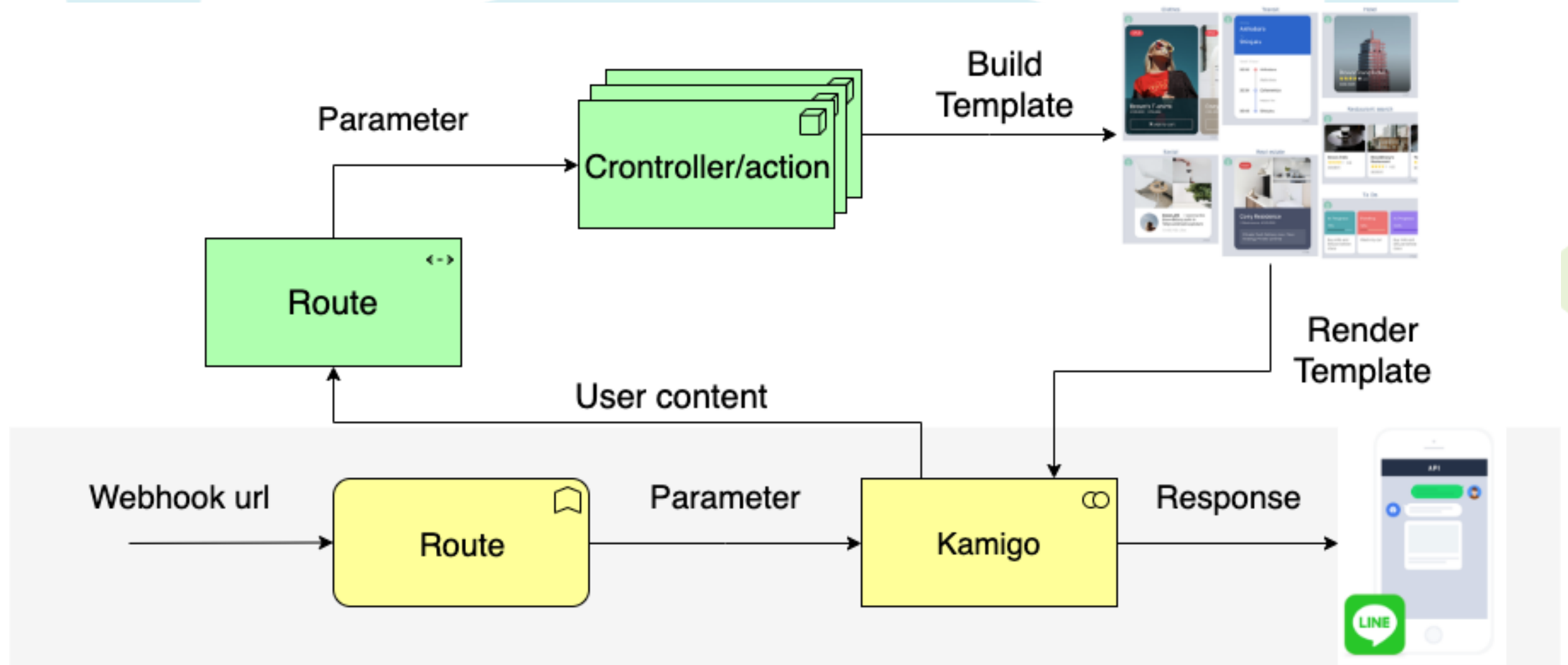
但我只用來開發 Chatbot !!

因為每次的搜尋都會確認用戶身份
(其實只是還沒實作在 Web)

設計理念: https://github.com/etrex/kamigo/blob/master/doc/01_intro.md

# 一般 Webhook 做法



參考

# Kamigo 做法



参考

為什麼選用 Kamigo？

絕對不會說我是米粉

# 帶來的好處

- 一開始就只是覺得很厲害

- 一個 Event 交給一個 Route(Controller)

- 寫起來很像在寫 Rails

- 組 flex message 不用看一堆括號

- 支援一些簡單的語意理解規則

- 懶

接著來看一下路由可以怎麼設計

```ruby
Rails.application.routes.draw do

  root to: "practices#index"
  get '首頁', to: "practices#index"
  get "上次(*train)", to: "practices#last_record"
  get "今日訓練", to: "practices#index_details"
  get "過程(*any_type)", to: "practices#index_station"
  resources :practices do
    collection do
      post :search
      get :search
      post :history
      get :history
      get :index_details
      get :index_station
      get :last_record
    end
  end

  resources :users do
    collection do
      get :line_login
      post :line_login
      get :auth
    end
  end
  get 'liff_entry', to: 'liff#entry'
  post 'liff_route', to: 'liff#route'
  get "(*other)", to: "others#index"
end
```

12:21

POST /practices/search
{"trip":
{"departing(1i)":"2019","departing(2i)":"10","departing(3i)":"14"}}

已讀
下午 12:21

| 首頁 | 2019-10-14 |
|---|---|
| 訓練項目 | 重量 (kg) | |
| 史密斯平胸 | 50.0 | ✖ |
| 史密斯上胸 | 40.0 | ✖ |
| 啞鈴夾胸 | 16.0 | ✖ |
| Cable三… | 18.7 | ✖ |
| Cable三頭 | 6.8 | ✖ |
| Cable平胸 | 4.6 | ✖ |

總共訓練時間約：0 時 40 分 50 秒

| 單項成績 | 新增 | 首頁 | 查詢日期 | 上次訓練 |
|---|---|---|---|---|

▼
紀錄是我的本命

下午 12:21

POST /practices/search
{"trip":
{"departing(1i)":"2019","departing(2i)":"10","departing(3i)":"14"}}

已讀
下午 12:21

| 首頁 | 2019-10-14 |
|---|---|

| 訓練項目 | 重量 (kg) | |
|---|---|---|
| 史密斯平胸 | 50.0 | ✖ |
| 史密斯上胸 | 40.0 | ✖ |

```ruby
Rails.application.routes.draw do

  root to: "practices#index"
  get '首頁', to: "practices#index"
  get "上次(*train)", to: "practices#last_record"
  get "今日訓練", to: "practices#index_details"
  get "過程(*any_type)", to: "practices#index_station"
  resources :practices do
    collection do
      post :search
      get :search
      post :history
      get :history
      get :index_details
      get :index_station
      get :last_record
    end
  end

  resources :users do
    collection do
      get :line_login
      post :line_login
      get :auth
    end
  end
  get 'liff_entry', to: 'liff#entry'
  post 'liff_route', to: 'liff#route'
  get "(*other)", to: "others#index"
end
```

映射到對應的路由上

```ruby
Rails.application.routes.draw do

  root to: "practices#index"
  get '首頁', to: "practices#index"
  get "上次(*train)", to: "practices#last_record"
  get "今日訓練", to: "prac...dex_details"
  get "過程(*any_type)", t...ices#index_station"
  resource...
    collection do
      post :search
      get :search
      post :history
      get :history
      get :index_details
      get :index_station
      get :last_record
    end
  end

  resources :users do
    collection do
      get :line_login
      post :line_login
      get :auth
    end
  end
  get 'liff_entry', to: 'liff#entry'
  post 'liff_route', to: 'liff#route'
  get "(*other)", to: "others#index"
end
```

同理：
**GET /practices/search {xxxx}**
也會對應路由

支援模糊搜尋

# 範例

已讀
下午 1:18　上一次練了什麼?

| 首頁 | 2019-10-14 |
| --- | --- |
| **訓練項目** | **重量(kg)** |
| 史密斯平胸 | 50.0 ✖ |
| 史密斯上胸 | 40.0 ✖ |
| 啞鈴夾胸 | 16.0 ✖ |
| Cable三... | 18.7 ✖ |
| Cable三頭 | 6.8 ✖ |
| Cable平胸 | 4.6 ✖ |

下午 1:18

```ruby
Rails.application.routes.draw do

  root to: "practices#index"
  get '首頁', to: "practices#index"
  get "上次(*train)", to: "practices#last_record"
  get "今日訓練", to: "practices#index_details"
  get "過程(*any_type)", to: "practices#index_station"
  resources :practices do
    collection do
      post :search
      get :search
      post :history
      get :history
      get :index_details
      get :index_station
      get :last_record
    end
  end

  resources :users do
    collection do
      get :line_login
      post :line_login
      get :auth
    end
  end
  get 'liff_entry', to: 'liff#entry'
  post 'liff_route', to: 'liff#route'
  get "(*other)", to: "others#index"
end
```

```ruby
Rails.application.routes.draw do

  root to: "practices#index"
  get '首頁', to: "practices#index"
  get "上次(*train)", to: "practices#last_record"
  get "今日訓練", to: "practices#index_details"
  get ...                                    ...ation"
  re...
      coll...
          po...
          get :search
          post :history
          get :history
          get :index_details
          get :index_station
          get :last_record
      end
  end

  resources :users do
    collection do
      get :line_login
      post :line_login
      get :auth
    end
  end
  get 'liff_entry', to: 'liff#entry'
  post 'liff_route', to: 'liff#route'
  get "(*other)", to: "others#index"
end
```

其實我就只判斷 "上次xxx"
其他的交給 Rails 幫忙

```ruby
1   Rails.application.routes.draw do
2
3     root to: "practices#index"
4     get '首頁', to: "practices#index"
5     get "上次(*train)", to: "practices#last_record"
6     get "今日訓練", to: "practices#index_details"
7     get "過程(*any_type)", to: "practices#index_station"
8     resources :practices do
9       collection do
10        post :search
11        get :search
12        post :history
13        get :history
14        get :index_details
15        get :index_station
16        get :last_record
17      end
18    end
19
20    resour
21      collection do
22        get :line_login
23        post :line_login
24        get :auth
25      end
26    end
27    get 'liff_entry', to: 'liff entry'
28    post 'liff_route', to: 'liff#route'
29    get "(*other)", to: "others#index"
30  end
```

到最後沒人要的文字就會在這

# 所以

- 只要符合 上次xxx 接送往對應 Controller

- 還有可以判斷中文的路由！！

- 不用再寫一堆 if else，全讓路由去處理

- Rails 寫起來好舒服

好舒服呀～

Kamiflex

（Flex Message）

# 這是你看到的訊息

| 首頁 | | 2019-10-14 |
|------|------|------|
| 訓練項目 | 重量(kg) | |
| 史密斯平胸 | 50.0 | ✕ |
| 史密斯上胸 | 40.0 | ✕ |
| 啞鈴夾胸 | 16.0 | ✕ |
| Cable三... | 18.7 | ✕ |
| Cable三頭 | 6.8 | ✕ |
| Cable平胸 | 4.6 | ✕ |

下午 9:57

```json
{
    "type": "flex",
    "altText": "this is a flex message",
    "contents": {
        "type": "bubble",
        "body": {
            "type": "box",
            "layout": "vertical",
            "contents": [
                {
                    "type": "box",
                    "layout": "horizontal",
                    "contents": [
                        {
                            "type": "text",
                            "text": "首頁",
                            "flex": 0,
                            "action": {
                                "type": "message",
                                "label": "首頁",
                                "text": "首頁"
                            },
                            "color": "#212121"
                        },
                        {
                            "type": "text",
                            "text": "2019-10-28",
                            "align": "end"
                        }
                    ],
                    "margin": "xs"
    ...以下省略三百行
                }
            ]
```

這是原本的樣子

# 我以前就是這樣刻 JSON

# 這是我寫的內容

```erb
<%= raw(Kamiflex.build(self) do
  bubble do
    body do
      body_header(flag, train_date)
      if @practices.present?
        vertical_box spacing: "sm", margin: "lg" do
          horizontal_box margin: "lg" do
            text "訓練項目", weight: "bold"
            text "重量(kg)", align: "end"
            text " ", align: "end"
          end
          horizontal_box @practices, margin: "lg" do |practice|
            text "#{practice.bar_type}#{practice.moving}", action: message_action("GET /practices/#{practice.id}"),
              color: "#241663"
            text practice.weight.blank? ? "-" : practice.weight.to_s, action: message_action("GET /practices/
              #{practice.id}"), align: "end", margin: "xs", color: "#6e2142"
            text "❌", align: "end", action: message_action("DELETE /practices/#{practice.id}")
          end
          if @total.present?
            separator
            horizontal_box margin: "lg" do
              text "總共訓練時間約： #{@total[0]}時 #{@total[1]}分 #{@total[2]}秒", margin: "xs", color: "#6e2142"
            end
          end
        end
      else
        none_practice_message
      end # @practices present condition
    end # body
  end # bubble
end )%>
```

# Kamiflex 幫忙做的事

如此一來就可以用 Ruby 的風格去開發 Flex message

整個語意看起來就乾淨多了！！

好舒服呀~

接下來說一下 Kamiliff

# 這是我的 LIFF

line://app/1613867810-ymYpONB4?
liff_size=full&path=%2Fpractices%2Fnew

# 這是一般 LIFF 網址

line://app/1613867810-ymYpONB4?liff_size=full&path=%2Fpractices%2Fnew

# 後台的設定

| Name | LIFF URL | Size |
|------|----------|------|
| **compact** | line://app/1613867810-nB7EAG2g | Compact |
| **tall** | line://app/1613867810-rowDxVBE | Tall |
| **liff_test** | line://app/1613867810-XzpPGJWw | Full |
| **full** | line://app/1613867810-ymYpONB4 | Full |

Chatbot TW

# Kamiliff 加的東西

"line://app/1613867810-ymYpONB4?
liff_size=full&path=%2Fpractices%2Fnew"

# LIFF 尺寸控制

"line://app/1613867810-ymYpONB4?
liff_size=full&path=%2Fpractices%2Fnew"

LIFF 的三種尺寸在這裡設定

# Url encode

"line://app/1613867810-ymYpONB4?
liff_size=full&path=%2Fpractices%2Fnew"

斜線                              斜線

# 路由控制

"line://app/1613867810-ymYpONB4?
liff_size=full&path=%2Fpractices%2Fnew"

/practices/new

# 就是這個啦！

| 動詞 | Prefix | 路徑 | Controller |
|------|--------|------|------------|
| GET | users | /users | UsersController |
| POST | users | /users | UsersController |
| GET | new_user | /users/new | UsersController |
| GET | edit_user | /users/:id/edit | UsersController |
| GET | user | /users/:id | UsersController |
| PATCH | user | /users/:id | UsersController |
| PUT | user | /users/:id | UsersController |
| DELETE | user | /users/:id | UsersController |

(示意圖)

# 需要 LIFF 的 Route

| 動詞 | Prefix | 路徑 | Controller |
|------|--------|------|------------|
| GET | users | /users | UsersController |
| POST | users | /users | UsersController |
| GET | new_user | /users/new | UsersController |
| GET | edit_user | /users/:id/edit | UsersController |
| GET | user | /users/:id | UsersController |
| PATCH | user | /users/:id | UsersController |
| PUT | user | /users/:id | UsersController |
| DELETE | user | /users/:id | UsersController |

(示意圖)

# 一句話 ~~Hack~~ 產生 LIFF path

liff_path(path: new_practice_path, liff_size: 'full')

# 一句話 ~~Hack~~ 產生 LIFF path

liff_path(path: new_practice_path, liff_size: 'full')

**Kamiliff 提供的方法**

# 後台只要設定三個就好

| LIFF URL | Size |
|----------|------|
| line://app/1566713654-5ByzX7Wz | Tall |
| line://app/1566713654-A8y60ov6 | Compact |
| line://app/1566713654-m5pgO6xg | Full |

Chatbot TW

# 其他都讓 Kamiliff 去處理

| LIFF URL | Size |
|---|---|
| line://app/1566713654-5ByzX7Wz | Tall |
| line://app/1566713654-A8y60ov6 | Compact |
| line://app/1566713654-m5pgO6xg | Full |

Chatbot TW

好舒服呀～

# 題外話

Kamigo 現在使用的還是 v1
但是 LIFF 已經有 v2 囉！

# 題外話

結尾時間
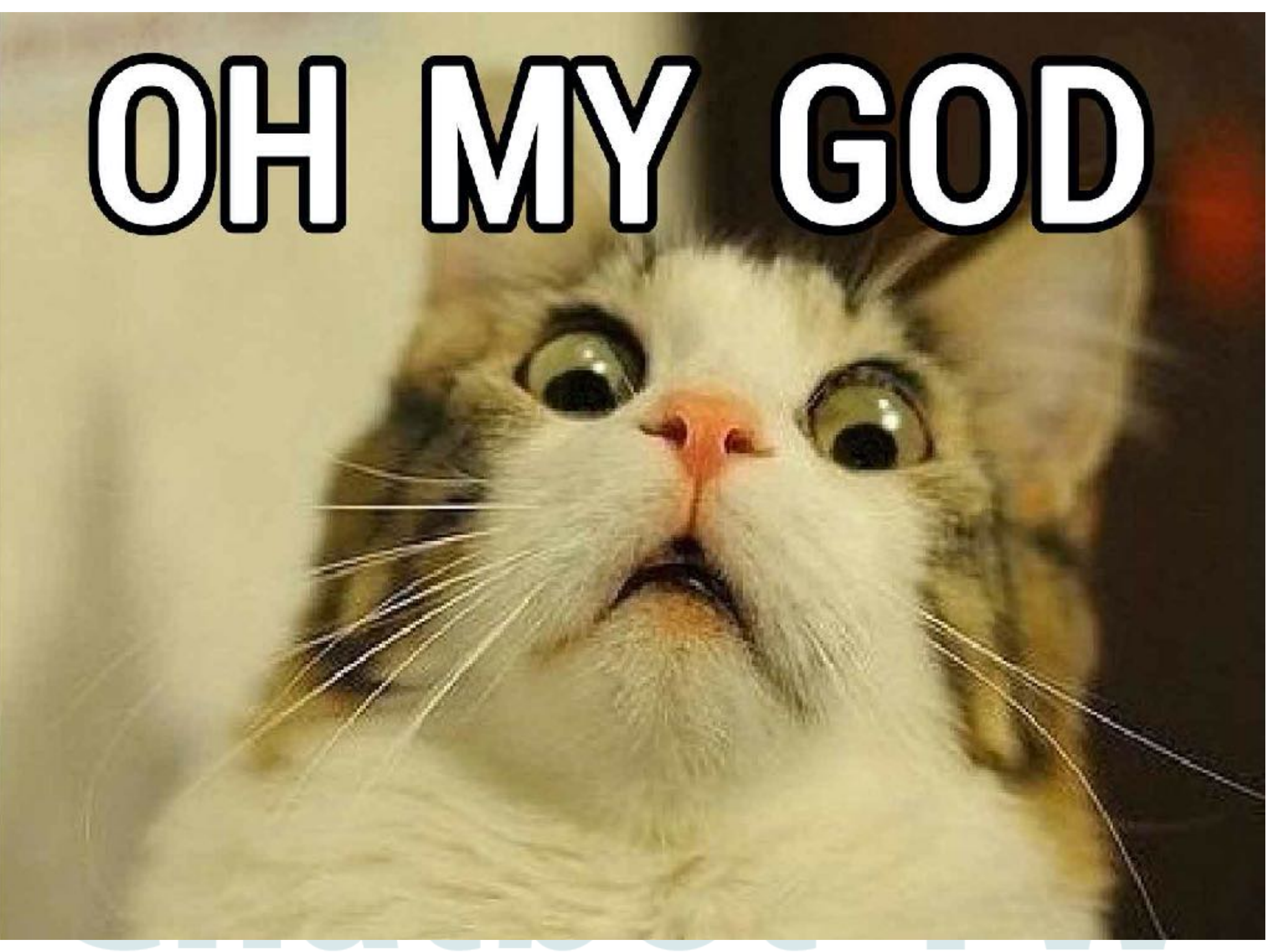
最後看看我以前寫的東西

```ruby
if text[0] == "sign"
  res = HTTParty.get('https://api.twitch.tv/kraken/users', query: {'login': "#{text[1]}"}, headers: {'Authorization': 'OAu
  user = res.parsed_response["users"][0]
  twitch.update_attributes({
    _id: user['_id'],
    display_name: user['display_name'],
    name: user['name'],
    created_at: user['created_at']})
  msg = "_id: #{user['_id']}\ndisplay_name: #{user['display_name']}\nname: #{user['name']}\n建立時間: #{user['created_at']}\
  message = { "type": "text", "text": msg }
  reply_to_line(message)
elsif text[0] == "follow"
  unless twitch.name.nil?
    res = HTTParty.get("https://api.twitch.tv/kraken/users/#{twitch['_id']}/follows/channels", headers: {'Authorization':
    channel = ""
    r = res.parsed_response
    r["follows"].each_with_index { |val, idx|
      idx < r["follows"].length - 1 ? channel = "#{channel}#{val['channel']['name']}," : channel = "#{channel}#{val['chan
    }
    streams = HTTParty.get("https://api.twitch.tv/kraken/streams/", query: {'channel': "#{channel}"}, headers: { 'Client-I
    total = []
    total_mobile = []
    st = streams.parsed_response["streams"]
    if st.blank?
      total = "現在追隨的實況主都沒開嘀^_^"
    else
      st.each do |s|
```

# OH MY GOD

讓我能無腦的開發

專心在機器人身上

如果寫的語言沒有怎麼辦？

沒有就自己寫一個！！

沒有就自己寫一個！！

沒有就自己寫一個！！

或是問剛剛的大大 🤩

或卡米哥 (出賣)

# 工商服務時間

徵志工、講者！

歡迎休息時間來詢問了解！

Chatbot TW

謝謝大家！！