5.2 LINE Bot API

LINE Bot SDK 提供了許多 API 讓設計者以程式與使用者互動，其中最常使用的就是收到使用者訊息後給予適當的回應。

5.2.1 回應訊息基本語法

當使用者傳送訊息給 LINE Bot 時，會觸發 MessageEvent 事件，此處僅處理收到的文字訊息，建立路由的語法為：

```
@handler.add(MessageEvent, message=TextMessage)
```

「message=TextMessage」表示收到的是文字訊息：即只有收到的是文字訊息才由此路由處理。

接著建立處理路由的函式：

```
def 函式名稱(event):
```

參數 event 包含傳回的各項訊息。例如建立的函式名稱為 handle_message：

```
def handle_message(event):
```

通常文字處理程式的第一步是取得使用者傳送的文字，語法為：

```
傳送文字變數 = event.message.text
```

例如取得使用者傳送的文字存於 mtext 變數中。

```
mtext = event.message.text
```

接著根據使用者傳送的文字做適當的處理。綜合以上步驟，LINE Bot 互動功能的基本語法為：

```
@handler.add(MessageEvent, message=TextMessage)
def handle_message(event):
    mtext = event.message.text
        if mtext == 傳送文字一:
            處理程式一
        if mtext == 傳送文字二:
            處理程式二
       ……
```


碁峰 www.gotop.com.tw

5.2.2 回傳文字訊息

回傳訊息 (reply_message) 的種類有 Text (文字)、Image (圖片)、Location (位置)、Sticker (貼圖)、Audio (聲音)、Video (影片) 及 Template (樣板) 等。

回傳訊息的語法為：

```
line_bot_api.reply_message(event.reply_token, 訊息種類 )
```

上述語法的「訊息種類」由訊息命令及參數組成，語法為：

```
訊息命令 ( 參數一 = 值一 , 參數二 = 值二 , ...)
```

最簡單的回傳訊息是文字，回傳文字訊息的訊息命令為 TextSendMessage，回傳文字訊息的語法為：

```
line_bot_api.reply_message(event.reply_token,
  TextSendMessage(text= 文字訊息內容 ))
```

通常訊息種類的參數不只一個，導致回傳訊息程式碼相當長，會降低程式可讀性，此時可改用下面語法：

```
訊息變數 = 訊息命令 (
     參數一 = 值一 ,
     參數二 = 值二 ,
     ...
)
line_bot_api.reply_message(event.reply_token, 訊息變數 )
```

例如上面的回傳文字訊息程式為：

```
message = TextSendMessage(
    text = 文字訊息內容
)
line_bot_api.reply_message(event.reply_token,message)
```


碁峰 www.gotop.com.tw

5.2.3 建立回應訊息 LINE Bot

建立一個具有互動功能 LINE Bot 的步驟為：

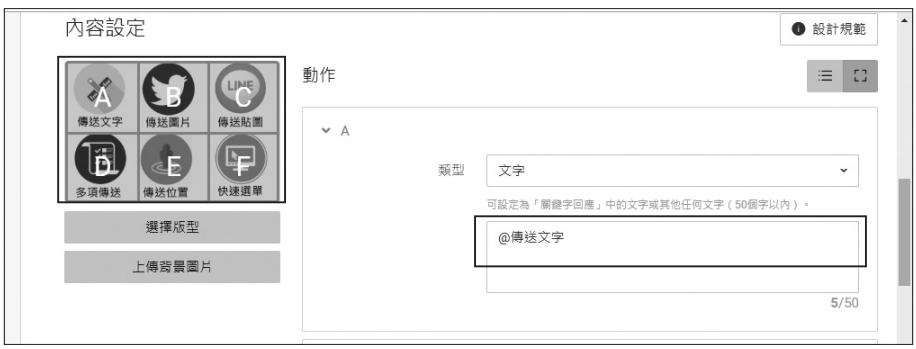
1. 在 Line 開發者網站建立一個 Messaging API Channel。
2. 建立一個建立 Flask 程式，並於程式中撰寫互動程式碼。
3. 執行 Flask 程式可啟動本機伺服器，再啟動 ngrok 伺服器，設定 Messaging API Channel 的 Webhook URL 為 ngrok 伺服器網址。

由於建立 LINE Bot 過程略顯繁複，此處將本節所有回應功能置於同一個 LINE Bot 中，以圖文選單來執行各小節的回應功能。

建立 Messaging API Channel

參考前一章操作，建立 ehappyFunc1 LINE Bot，接著加入圖文選單：版型使用「大型」的第一個版型 (六個項目)，圖形請上傳本章範例 <media/func1.png>，六個項目的類型皆選擇「文字」，傳送的文字分別設定為 @ 傳送文字、@ 傳送圖片、@ 傳送貼圖、@ 多項傳送、@ 傳送位置、@ 快速選單。

通常由圖文選單傳送的文字會在前面加上一個特殊字元做為區別，以與使用者自行輸入的文字做為區分，此處在傳送文字前面加上「@」。



建立 Flask 程式

接著建立 Flask 程式 <linebotFunc1.py>，讓 LINE Bot 回應使用者點選圖文選單的各項功能：根據使用者傳送的文字訊息進行處理，先列出處理「@ 傳送文字」的程式碼。

碁峰 www.gotop.com.tw

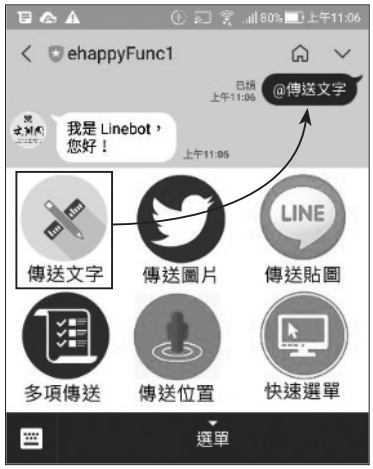
程式碼：linebotFunc1.py

```
...
 7 from linebot.models import MessageEvent, TextMessage,
      TextSendMessage, ImageSendMessage, StickerSendMessage,
      LocationSendMessage, QuickReply, QuickReplyButton, MessageAction
...
22 @handler.add(MessageEvent, message=TextMessage)
23 def handle_message(event):
24     mtext = event.message.text
25     if mtext == '@傳送文字':
26         try:
27             message = TextSendMessage(
28                 text = "我是 Linebot，\n您好！"
29             )
30             line_bot_api.reply_message(event.reply_token,message)
31         except:
32             line_bot_api.reply_message(event.reply_token,
                   TextSendMessage(text='發生錯誤！'))
...
```

程式說明

- 7　匯入各互動功能模組。
- 22-23　處理輸入文字訊息。
- 24　讀取使用者輸入的文字（由圖文選單產生）。
- 25　如果使用者傳送的文字訊息為「@ 傳送文字」。
- 26-30　LINE Bot 回傳文字訊息。
- 31-32　LINE Bot 發生錯誤時回傳訊息。

啟動本機及 ngrok 伺服器，然後將 LINE Bot 的 Webhook URL 設為 ngrok 伺服器的 https 伺服器網址。手機加入 ehappyFunc1 LINE Bot 為朋友，開啟 ehappyFunc1 LINE Bot 後點選「傳送文字」圖示的執行結果：



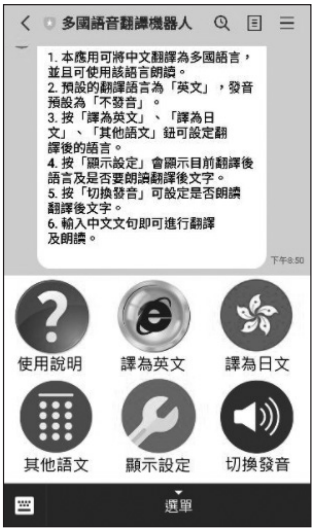
碁峰 www.gotop.com.tw

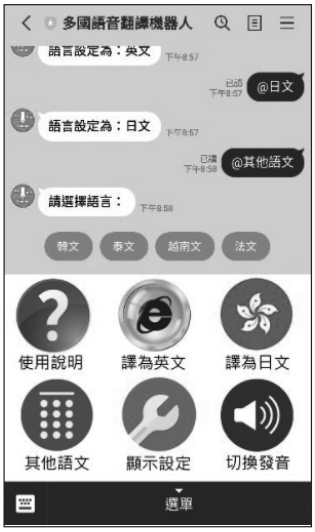
11.1 專題方向

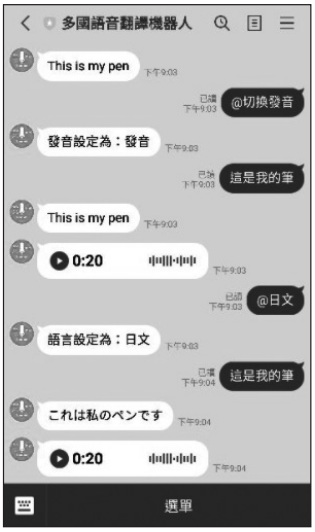
到國外旅行時，由於語言不通，如何與當地人溝通是一大問題，本專題利用行動裝置可播放聲音的特性，當使用者輸入文句，就先將其翻譯為指定語言的文句，再利用 Google 語音 API 雲端服務轉換為語音播放，輕鬆解決不同語言溝通的問題。

專題檢視

- 「使用說明」功能讓使用者了解本專題應用程式的使用方法。
- 「譯為英文」、「譯為日文」功能分別設定翻譯後語言為英文、日文。預設的翻譯語言為「英文」。
- 「其他語文」功能可以選擇韓文、泰文、越南文或法文。
- 「顯示設定」功能會顯示目前翻譯後語言及是否要朗讀翻譯後文字。
- 「切換發音」功能會改變目前發音狀態：若目前會朗讀文字則改為不朗讀文字，若目前不會朗讀文字則改為朗讀文字。
- 使用者直接輸入文字傳送，系統會執行翻譯，然後再視「發音狀態」的設定值決定是否轉換為語音。







碁峰 www.gotop.com.tw

11.2 關鍵技術

本專題使用 translate 模組將中文文句翻譯為其他語言，再呼叫 google 語音 API 將文句轉換為語音朗讀。

11.2.1 Google 語音 API

Google 提供許多雲端服務，文字轉語音 (TTS) 是相當受歡迎的服務之一。Google 雲端文字轉語音服務的網址為「https://google-translate-proxy.herokuapp.com/api/tts」，其參數有三個：

- **query**：要轉語音的文字字串。
- **language**：轉出語音的語言代碼。下表為常用的語言代碼：

語言	代碼	語言	代碼	語言	代碼
繁體中文	zh-Hant	簡體中文	zh-Hans	希臘文	el
英文	en	日文	ja	韓文	ko
法文	fr	泰文	th	荷蘭文	nl
德文	de	越南文	vi	西班牙文	es

- **speed**：轉出語音的發音速度，其值在 0.2 到 1 之間。

文字轉語音服務的傳回值是一個網址，例如將「今天天氣很好」輸出的語音存於 returl 變數的語法為：

```
returl = 'https://google-translate-proxy.herokuapp.com/api/tts?
    query=今天天氣很好&language=zh-Hant'
```

在 LINE Bot 中可用 AudioSendMessage 回覆訊息方式播放轉換後的語音。

例如在 LINE Bot 中設定以「####」開頭的輸入是將文字轉換為語音功能，程式碼為：

```
……
 1 from urllib.parse import quote
……
 2 @handler.add(MessageEvent, message=TextMessage)
```


碁峰 www.gotop.com.tw

11.3 實戰：多國語音翻譯機器人

出國旅遊時，不論食衣住行都會碰到語言溝通的問題。現在科技如此發達，只要手機有「多國語音翻譯機器人」，無論身處哪一個國家，只要輸入本國文字就能以該國語言讀出，溝通無障礙，開心出國玩吧！

本專題為簡化程式，輸入的語言限定為繁體中文。

11.3.1 資料表結構

在 LINE 開發者頁面建立「多國語音翻譯機器人」LINE Bot，加入圖文選單：版型使用「大型」的第一個版型 (六個項目)，圖形上傳本章範例 <media/translate.png>，所有項目類型皆選擇「文字」，傳送的文字分別設定為 @ 使用說明、@ 英文、@ 日文、@ 其他語文、@ 顯示設定、@ 切換發音。接著建立 Flask 程式 <linebotTranslate.py>，讓 LINE Bot 回應使用者點選圖文選單的各項功能。

如果尚未進行前一節操作，則需先安裝本節所需模組：

```
pip install translate==3.5.0
```

本章使用的資料表名稱為 setting，包含三個欄位：uid 欄位記錄使用者的 LINE Id，lang 欄位記錄輸出的語言，sound 欄位記錄是否要讀出語音。

執行 **程式集** / **PostgreSQL 13** / **pgAdmin 4** 開啟資料庫管理程式，在 **Databases** 按滑鼠右鍵，於快顯功能表點選 **Create** / **Database**。**Database** 欄位輸入資料庫名稱「translate」，**Owner** 欄位選擇管理者名稱 admin，按 **Save** 鈕完成建立名稱為 translate 的資料庫。

建立資料表的程式碼為：

程式碼：createtable.py

```
……
app.config['SQLALCHEMY_DATABASE_URI'] = 'postgresql://
      管理者名稱：管理者密碼@127.0.0.1:5432/translate'
db = SQLAlchemy(app)

@app.route('/')
```


碁峰 www.gotop.com.tw

```
 9 def index():
10     sql = """
11     CREATE TABLE setting (
12     id serial NOT NULL,
13     uid character varying(50) NOT NULL,
14     lang character varying(10) NOT NULL,
15     sound character varying(10) NOT NULL,
16     PRIMARY KEY (id))
17     """
18     db.engine.execute(sql)
19     return "資料表建立成功！"
……
```

程式說明

- 5-6　連接資料庫。
- 10-18　建立 setting 資料表。
- 13　「uid」欄位存使用者 LINE Id。
- 14　「lang」欄位儲存輸出的語言。
- 15　「sound」欄位儲存是否要讀出語音。

執行程式後開啟瀏覽器，網址列輸入「http://127.0.0.1:5000/」，即可見到「資料表建立成功！」。在資料庫管理頁面 **invoice** / **Schemas** / **public** / **Tables** 中可見到新建立的 translate 資料表。

LINE Bot 應用程式的特性是可能多人同時使用，因此必須將使用者的 LINE Id 及設定狀態 (翻譯語言及是否發音) 寫入資料庫，每次使用者傳送訊息時，再根據使用者的 LINE Id 資料庫讀出使用者設定狀態，這樣使用者的設定狀態才不會被其他使用者覆蓋。

碁峰 www.gotop.com.tw

11.3.2 「使用說明」功能

「使用說明」利用回覆文字訊息顯示本專題的使用方法。關於「使用說明」的程式碼為：

程式碼：**linebotTranslate.py**

```
……
 9 from translate import Translator
10 from urllib.parse import quote
11 from urllib.parse import parse_qsl
……
26 app.config['SQLALCHEMY_DATABASE_URI'] = 'postgresql://admin:123456
                                          @127.0.0.1:5432/translate'
27 db = SQLAlchemy(app)
28
29 @handler.add(MessageEvent, message=TextMessage)
30 def handle_message(event):
31     userid, lang, sound = readData(event)  # 讀取原有設定
32     mtext = event.message.text
33     if mtext == '@使用說明':
34         showUse(event)
……
60 def readData(event):  # 讀取使用者 id, 語言及發音設定
61     userid = event.source.user_id
62     sql_cmd = "select * from setting where uid='" + userid + "'"
63     query_data = db.engine.execute(sql_cmd)
64     datalist = list(query_data)
65     if len(datalist) == 0:
66         sql_cmd = "insert into setting (uid, lang, sound)
               values('" + userid + "', 'en', 'no');"
67         db.engine.execute(sql_cmd)
68         lang = 'en'
69         sound = 'no'
70     else:
71         lang = datalist[0][2]
72         sound = datalist[0][3]
73     return userid, lang, sound
74
```


碁峰 www.gotop.com.tw

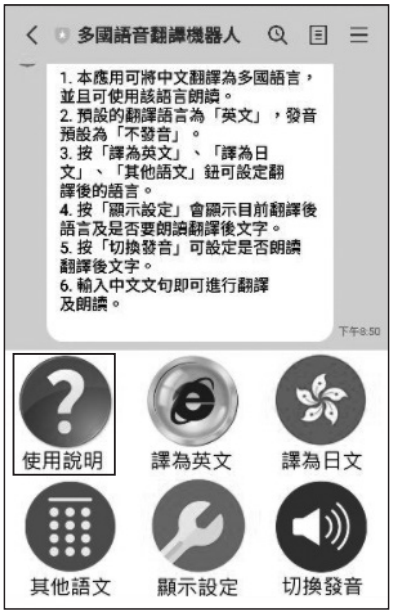
```
75 def showUse(event):
76     try:
77         text1 = '''
78 1. 本應用可將中文翻譯為多國語言，並且可使用該語言朗讀。
79 2. 預設的翻譯語言為「英文」，發音預設為「不發音」。
80 3. 按「譯為英文」、「譯為日文」、「其他語文」鈕可設定翻譯後的語言。
81 4. 按「顯示設定」會顯示目前翻譯後語言及是否要朗讀翻譯後文字。
82 5. 按「切換發音」可設定是否朗讀翻譯後文字。
83 6. 輸入中文文句即可進行翻譯及朗讀。
84                 '''
85         message = TextSendMessage(
86             text = text1
87         )
88         line_bot_api.reply_message(event.reply_token,message)
89     except:
90         line_bot_api.reply_message(event.reply_token,
               TextSendMessage(text=' 發生錯誤！'))
……
```

程式說明

- 9-11　匯入模組。
- 26-27　連接 translate 資料庫。
- 31　程式開始就執行 readData 函式：若 LINE Id 不存在就寫入資料庫，若 LINE Id 存在就由資料庫讀取該使用者的設定。
- 32-34　使用者按圖文選單「使用說明」就執行 sendUse 函式。
- 60-73　讀取使用者設定狀態的函式。
- 61　取得使用者 LINE Id。
- 62-64　檢查使用者 LINE Id 是否存在。
- 65-67　若資料表中無此使用者 LINE Id，就將使用者 LINE Id、輸出語言值「en」、是否讀出語音值「no」寫入資料表。
- 68-69　設定輸出語言值為「en」、是否讀出語音值為「no」。
- 70-72　若使用者 LINE Id 存在，就讀取輸出語言值及是否讀出語音值。
- 75-90　顯示使用說明的函式。
- 77-84　使用「'''」方式定義文字字串。
- 85-88　顯示文字訊息。

碁峰 www.gotop.com.tw

開啟本機及 ngrok 伺服器，點選 **使用說明** 的執行結果：



11.3.3 「譯為英文」及「譯為日文」功能

這兩個功能共用 setLang 函式，函式的第二個參數：點選 **譯為英文** 傳送「en」，點選 **譯為日文** 傳送「ja」。程式碼為：

程式碼：linebotTranslate.py（續）

```
……
    elif mtext == '@英文':
        setLang(event, 'en', sound, userid)

    elif mtext == '@日文':
        setLang(event, 'ja', sound, userid)
……
def setLang(event, lang, sound, userid):  # 設定翻譯語言
    try:
        sql_cmd = "update setting set lang='" + lang + "',
                  sound='" + sound + "' where uid='" + userid +"'"
        db.engine.execute(sql_cmd)
        message = TextSendMessage(
```


碁峰 www.gotop.com.tw