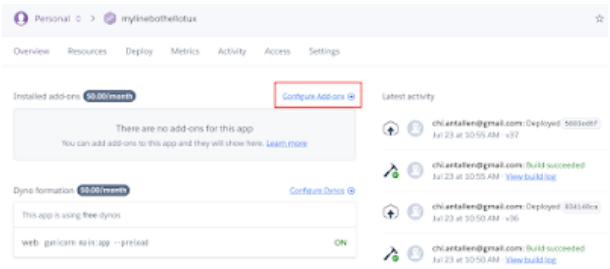
連結與使用資料庫

學習目標：

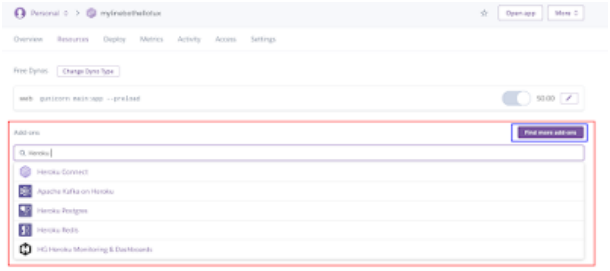
- 了解 Python 與資料庫連結方式！
- 設定 Line Bot 機器人讀寫資料庫！

Python 與資料庫連結

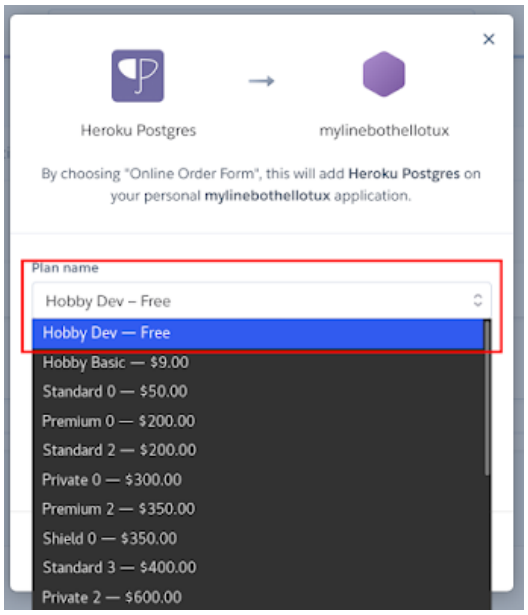
1. 在 Heroku 上，新增一個資料庫軟體：
    - 使用「Configure Add-ons」選項！



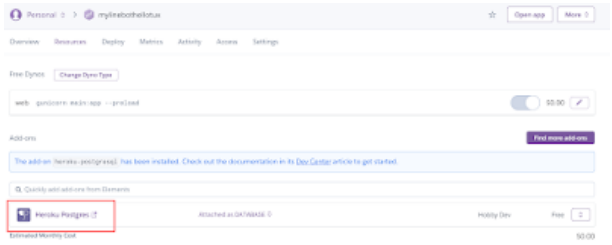
    - 選擇「Heroku Postgres」項目！



    - 「Plan Name」選用 Free 項目！再按下「Submit Order Form」！



- 按下「Heroku Postgres」可查看使用情形！



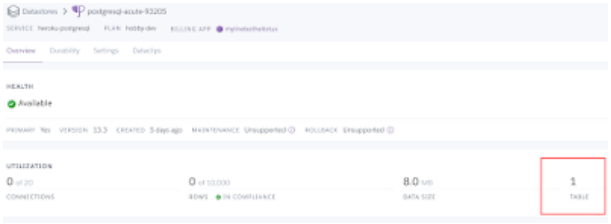
2. 使用文字介面視窗，安裝 psycopg2 套件，並進行資料庫的建立！

```
C:\workspace\LineBot> heroku login -i
C:\workspace\LineBot> pip install psycopg2
C:\workspace\LineBot> python3
>>> import os
>>> import psycopg2
>>> heroku_pgCLI = 'heroku config:get DATABASE_URL -a 你的APP名稱'
>>> DATABASE_URL = os.popen(heroku_pgCLI).read()[:-1]
>>> print(DATABASE_URL)
>>> connection_db=psycopg2.connect(DATABASE_URL, sslmode='require')
>>> cursor = connection_db.cursor()
>>>
(要注意一下網址！)
```

3. 建立一張表格，可以放入麵店的點餐單資料！

```
>>> SQL_cmd = '''CREATE TABLE menu(
... menu_id serial PRIMARY KEY,
... menuname VARCHAR (150) UNIQUE NOT NULL,
... menuprize Integer NOT NULL
... );'''
>>> cursor.execute(SQL_cmd)
>>> connection_db.commit()
>>> cursor.close()
>>> connection_db.close()
>>>
```

4. 查看一下 Heroku 內的資料！



5. 利用指令，查看資料庫表格相關訊息！

```
>>> import os
>>> import psycopg2
>>> heroku_pgCLI = 'heroku config:get DATABASE_URL -a 你的APP名稱'
>>> DATABASE_URL = os.popen(heroku_pgCLI).read()[:-1]
>>> connection_db=psycopg2.connect(DATABASE_URL, sslmode='require')
>>> cursor = connection_db.cursor()
>>> SQL_cmd = '''SELECT column_name, data_type FROM information_schema.columns WHERE table_name = 'menu';'''
>>> cursor.execute(SQL_cmd)
>>> connection_db.commit()
>>> data = []
>>> while True:
...   temp = cursor.fetchone()
...   if temp:
...      data.append(temp)
...   else:
...      break
...
>>> print(data)
[('menu_id', 'integer'), ('menuname', 'character varying'), ('menuprize', 'integer')]
>>> cursor.close()
>>> connection_db.close()
```

PS:刪除資料的方式：SQL_cmd = '"DROP TABLE IF EXISTS menu;"' ，再重複執行上列工作即可！

6. 將一筆資料輸入資料表內！

```
>>> import os
>>> import psycopg2
>>> heroku_pgCLI = 'heroku config:get DATABASE_URL -a 你的APP名稱'
>>> DATABASE_URL = os.popen(heroku_pgCLI).read()[:-1]
>>> connection_db=psycopg2.connect(DATABASE_URL, sslmode='require')
>>> cursor = connection_db.cursor()
>>> record = ('牛肉麵',170)
>>> table_columns = '(menuname,menuprize)'
>>> SQL_cmd = f"""INSERT INTO menu {table_columns} VALUES(%s,%s);"""
>>> cursor.execute(SQL_cmd,record)
>>> connection_db.commit()
>>> cursor.close()
>>> connection_db.close()
```

7. 將多筆資料輸入資料表內！

```
>>> import os
>>> import psycopg2
>>> heroku_pgCLI = 'heroku config:get DATABASE_URL -a 你的APP名稱'
>>> DATABASE_URL = os.popen(heroku_pgCLI).read()[:-1]
>>> connection_db=psycopg2.connect(DATABASE_URL, sslmode='require')
>>> cursor = connection_db.cursor()
>>> record = [('餛飩麵',100),('湯麵',80)]
>>> table_columns = '(menuname,menuprize)'
>>> SQL_cmd = f"""INSERT INTO menu {table_columns} VALUES(%s,%s);"""
>>> cursor.executemany(SQL_cmd,record)
>>> connection_db.commit()
>>> count = cursor.rowcount
>>> print(count," records have inserted!!")
>>> cursor.close()
>>> connection_db.close()
```

8. 列出資料庫的資料表內容

```
>>> import os
>>> import psycopg2
>>> heroku_pgCLI = 'heroku config:get DATABASE_URL -a 你的APP名稱'
>>> DATABASE_URL = os.popen(heroku_pgCLI).read()[:-1]
>>> connection_db=psycopg2.connect(DATABASE_URL, sslmode='require')
>>> cursor = connection_db.cursor()
>>> SQL_cmd = f"""SELECT * FROM menu ;"""
>>> cursor.execute(SQL_cmd)
>>> connection_db.commit()
>>> temp = cursor.fetchall()
>>> print(temp)
>>> count = cursor.rowcount
>>> print(count," records have geted!!")
>>> cursor.close()
>>> connection_db.close()
```

9. 有條件的列出資料庫的資料表內容：

```
>>> import os
>>> import psycopg2
>>> heroku_pgCLI = 'heroku config:get DATABASE_URL -a 你的APP名稱'
>>> DATABASE_URL = os.popen(heroku_pgCLI).read()[:-1]
>>> connection_db=psycopg2.connect(DATABASE_URL, sslmode='require')
>>> cursor = connection_db.cursor()
>>> prize = 100
>>> SQL_cmd = f"""SELECT * FROM menu WHERE menuprize < %s;"""
>>> cursor.execute(SQL_cmd,[prize])
>>> connection_db.commit()
>>> temp = cursor.fetchall()
>>> print(temp)
>>> count = cursor.rowcount
>>> print(count," records have geted!!")
>>> cursor.close()
>>> connection_db.close()
```

10. 更新資料庫的資料表內容：

```
>>> import os
>>> import psycopg2
>>> heroku_pgCLI = 'heroku config:get DATABASE_URL -a 你的APP名稱'
>>> DATABASE_URL = os.popen(heroku_pgCLI).read()[:-1]
>>> connection_db=psycopg2.connect(DATABASE_URL, sslmode='require')
>>> cursor = connection_db.cursor()
>>> name = "湯麵"
>>> prize = 50
>>> SQL_cmd = f"""UPDATE menu SET menuprize = %s WHERE menuname = %s;"""
>>> cursor.execute(SQL_cmd,(prize,name))
>>> connection_db.commit()
>>> count = cursor.rowcount
>>> print(count," records have updated!!")
>>> cursor.close()
>>> connection_db.close()
```

11. 刪除資料庫內的資料表內容："DELETE FROM menu WHERE menuname = ';"

Python 與資料庫連結

1. 修改 LineBot 專案的 requirements.txt：

```
Flask>=1.1.1
gunicorn>=20.0.4
line-bot-sdk>=1.15.0
APScheduler>=3.6.1
psycopg2>=2.0.0
```

2. 修改 LineBot 專案的程式碼：app/linebotmodules.py

```
import app.dataSQL import callData
(中間略過....)
# 靈活展現文字
```

```
@handler.add(MessageEvent, message=TextMessage)
def replyText(event):
    if event.source.user_id == "Uf4a596a6eb65eabf52c003ffe325a21d":
       reply = False
       if not reply:
           reply = callData.showData(event)
(接下來的程式碼，請先註解....)
```

3. 在 app 資料夾新增 dataSQL 資料夾！
4. 在資料夾 dataSQL 中，新增檔案 connectDB.py:

```
import os
import psycopg2

def showallMenu():
    DATABASE_URL = os.environ['DATABASE_URL']
    connection_db = psycopg2.connect(DATABASE_URL,sslmode='require')
    cursor = connection_db.cursor()
    SQL_cmd = f"""SELECT * FROM menu ;"""
    cursor.execute(SQL_cmd)
    raws = cursor.fetchall()
    data = []
    for raw in raws:
        data.append(raw)

    return data
```

5. 在資料夾 dataSQL 中，新增檔案 callData.py:

```
from app import line_bot_api
from linebot.models import TextSendMessage, ImageSendMessage, TemplateSendMessage, messages
from linebot.models import ImageCarouselTemplate, ImageCarouselColumn, URIAction
from app.dataSQL import connectDB

import random

def showData(event):
    if '菜單' in event.message.text:
        data = connectDB.showallMenu()
        print_text = ""
        for i in data:
            print_text += str(i[1])
            print_text += str(i[2])
            print_text += "\n"

        line_bot_api.reply_message(
            event.reply_token,
            TextSendMessage(text=print_text)
        )

        return True
    else:
        return False
```