**創建虛擬環境**

開啟終端機後，輸入以下指令：python -m venv 虛擬環境名稱

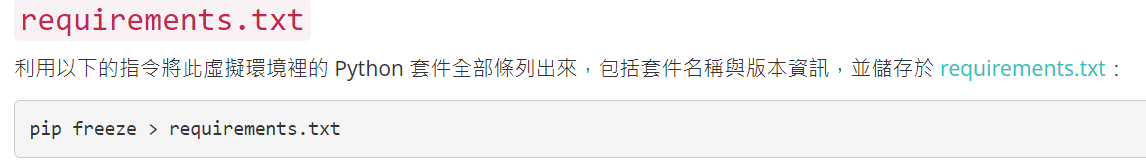
**切換虛擬環境**

開啟終端機後，輸入以下指令：虛擬環境名稱\Scripts\activate

###自己的:django\_venv\Scripts\activate

****

****

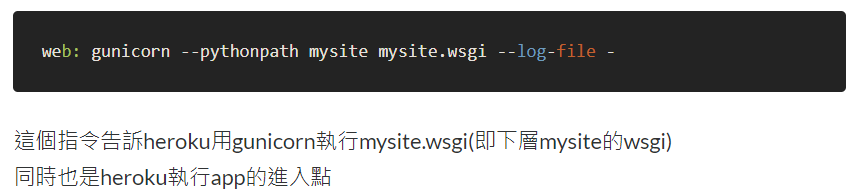
****

由於 Heroku 使用 [PostgreSQL](http://www.postgresql.org/) 資料庫，我們還需要手動在 requirements.txt 最後面加上  **psycopg2==2.9.1**

在網站根目錄(即manage.py的所在目錄)建立Procfile檔案

注意，沒有附檔名喔！而且P必須為大寫

內容輸入



**修改 wsgi.py**

heroku處理靜態檔案的方式與本機不相同  
而且為了讓heroku能夠透過wsgi與我們的網站溝通  
將原本的wsgi.py

import os

from django.core.wsgi import get\_wsgi\_application

os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'mysite.settings')

application = get\_wsgi\_application()

修改為

import os

from django.core.wsgi import get\_wsgi\_application

from dj\_static import Cling *# <- 加入*

os.environ.setdefault("DJANGO\_SETTINGS\_MODULE", "mysite.settings")

application = Cling(get\_wsgi\_application()) *# <- 修改*

**建立 production\_settings.py**

通常正式上線（production）時的環境會和開發時所做的settings.py設定有所不同  
而且我們希望網站上線時使用一個全新的、空白的資料庫  
以便與原本測試用的資料庫來做區分  
所以我們另外建立一個production\_settings.py，放在原本的settings.py旁邊

import dj\_database\_url

from .settings import \* *# 含入原本的settings.py所有設定*

*# heroku使用的資料庫為PostgreSQL，所以要修改資料庫設定*

DATABASES = {

'default': dj\_database\_url.config(),

}

STATIC\_ROOT = 'staticfiles' *# 設定網站正式上線時靜態檔案目錄位置*

SECURE\_PROXY\_SSL\_HEADER = ('HTTP\_X\_FORWARDED\_PROTO', 'https') *# 設定HTTP連線方式*

ALLOWED\_HOSTS = ['\*'] *# 讓所有的網域都能瀏覽本網站*

DEBUG = False *# 關閉除錯模式*

**建立 .gitignore**

為了要節省伺服器空間，我們不會把開發時使用的檔案，例如虛擬環境、本機資料庫及測試用檔案，一起部署上去  
因此，在網站根目錄建立一個.gitignore檔案，git在commit時會自動略過它們(後面會講)  
注意這個檔案就是以點.開頭，然後gitignore後面沒有其他的副檔名

.vscode

\*.pyc

\_\_pycache\_\_

db.sqlite3

Include

Lib

Scripts

tcl

Include、Lib、Scripts、tcl這些即為當初建立虛擬環境時所產生的檔案目錄

**新增新的 Heroku App**

接下來，我們需要新增一個可以上傳 repository 的地方：（如果你之前已經新增過，請直接跳到下一個步驟）

heroku create 自己app名稱

在本機初始化git並新建一個git儲存庫(repository)來存放和記錄網站檔案

git init

再將此git儲存庫與剛才你建立的heroku應用程式建立連結

heroku git:remote -a <你的heroku App name>

-a表示指定哪個應用程式

設定heroku使用production\_settings.py做為網站設定檔

heroku config:set DJANGO\_SETTINGS\_MODULE=mysite.production\_settings

將網站所有檔案加入git的追蹤

git add .

將所有追蹤的檔案加入git儲存庫，並將此次commit的動作命名為first commit

git commit -m "first commit"

commit好後，將檔案推送至heroku

git push heroku master

部署完成後，因為我們做的設定是使用全新的空白資料庫  
所以要同步資料庫並建立新的超級使用者

heroku run python manage.py migrate

heroku run python manage.py createsuperuser

最後讓heroku運行網站

heroku ps:scale web=1

如果忘記你的網站網址的話  
也可以直接執行

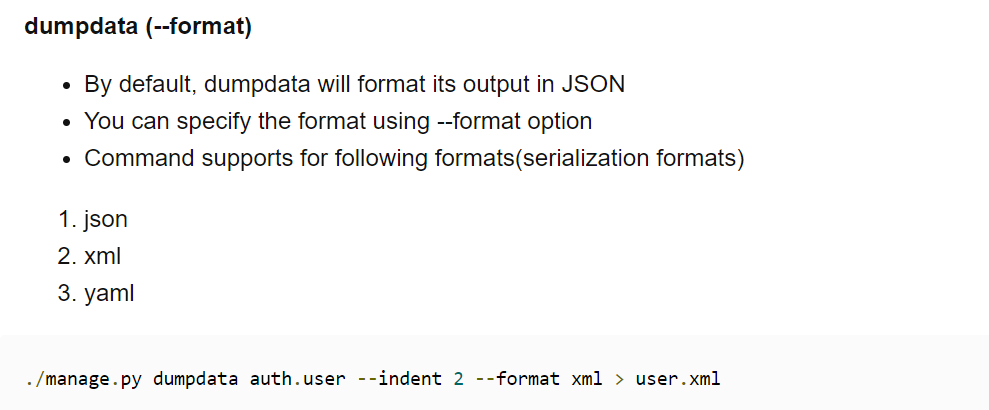
heroku open

**以上內容參考**

<https://ithelp.ithome.com.tw/articles/10212659?sc=rss.qu>

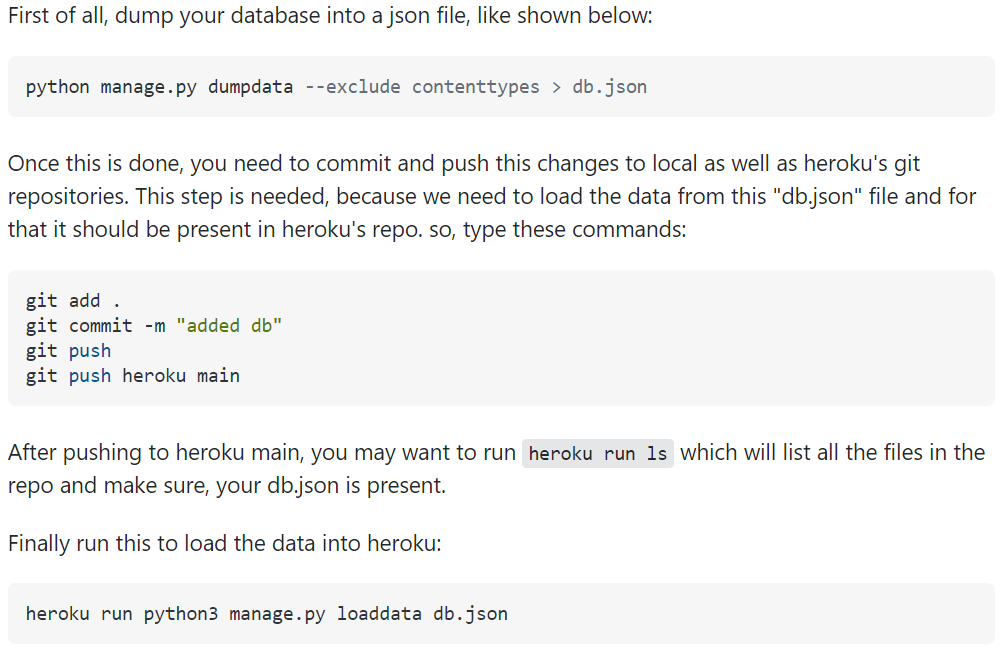
**Django dumpdata and loaddata**

<https://coderwall.com/p/mvsoyg/django-dumpdata-and-loaddata>



**How to transfer your local sqlite3 data to heroku?**

<https://stackoverflow.com/questions/61336258/how-to-transfer-your-local-sqlite3-data-to-heroku>



###git push heroku master