



django

Django

Django簡介

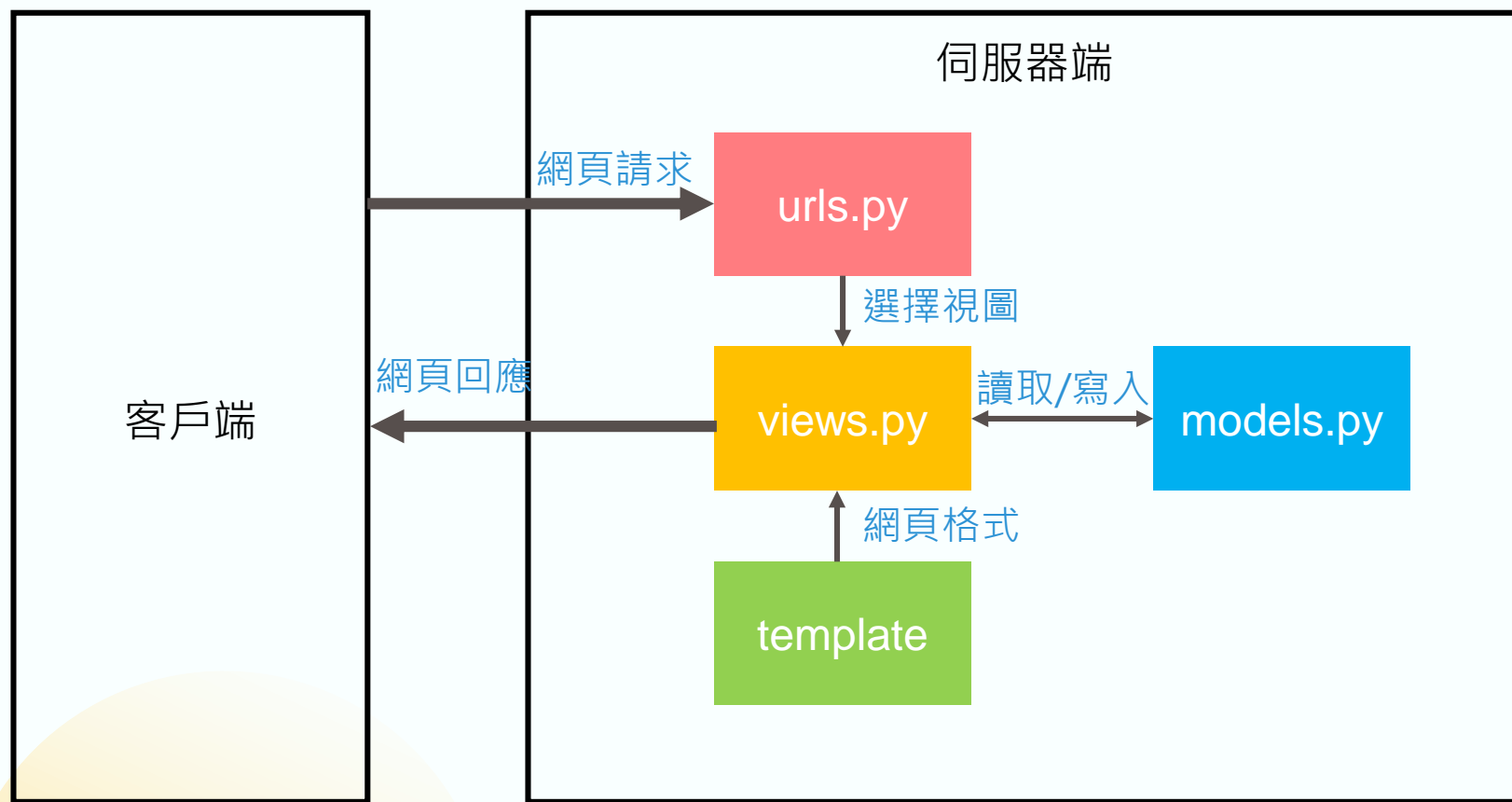
Django是一個開放原始碼的Web應用框架，由Python寫成
採用了MTV(model–template–views)的軟體設計模式
即模型（ Model ），視圖（ View ）和模板（ Template ）

django

MTV架構

- ◆ 目的是實現一種動態的程式設計，使後續對程式的修改和擴充功能簡化，並且使程式某一部分的重複利用成為可能
- ◆ 模型(model)：資料庫讀取、程式主要功能
- ◆ 模板(template)：將頁面展示給使用者看
- ◆ 視圖(view)：處理數據，並適當的調用model與template

MTV架構



MVC與MTV的差異

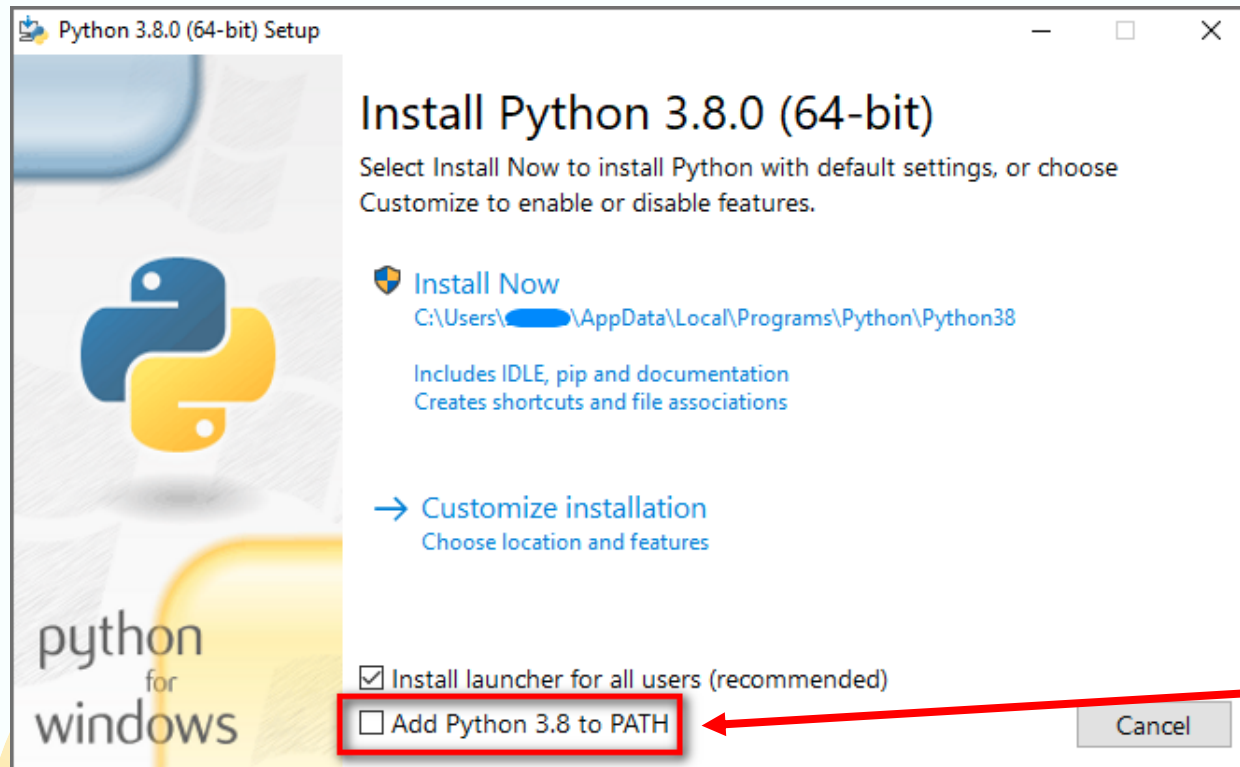
動態網頁的任務	MVC架構	MTV架構
資料庫互動與存取	M=model(模型)	M=model(模型)
網頁架構與樣式	V=view(視圖)	T=template(模版)
控制與整合	C=controller(控制)	V=view(視圖)

安裝django

Install Django

Step 1. 安裝python

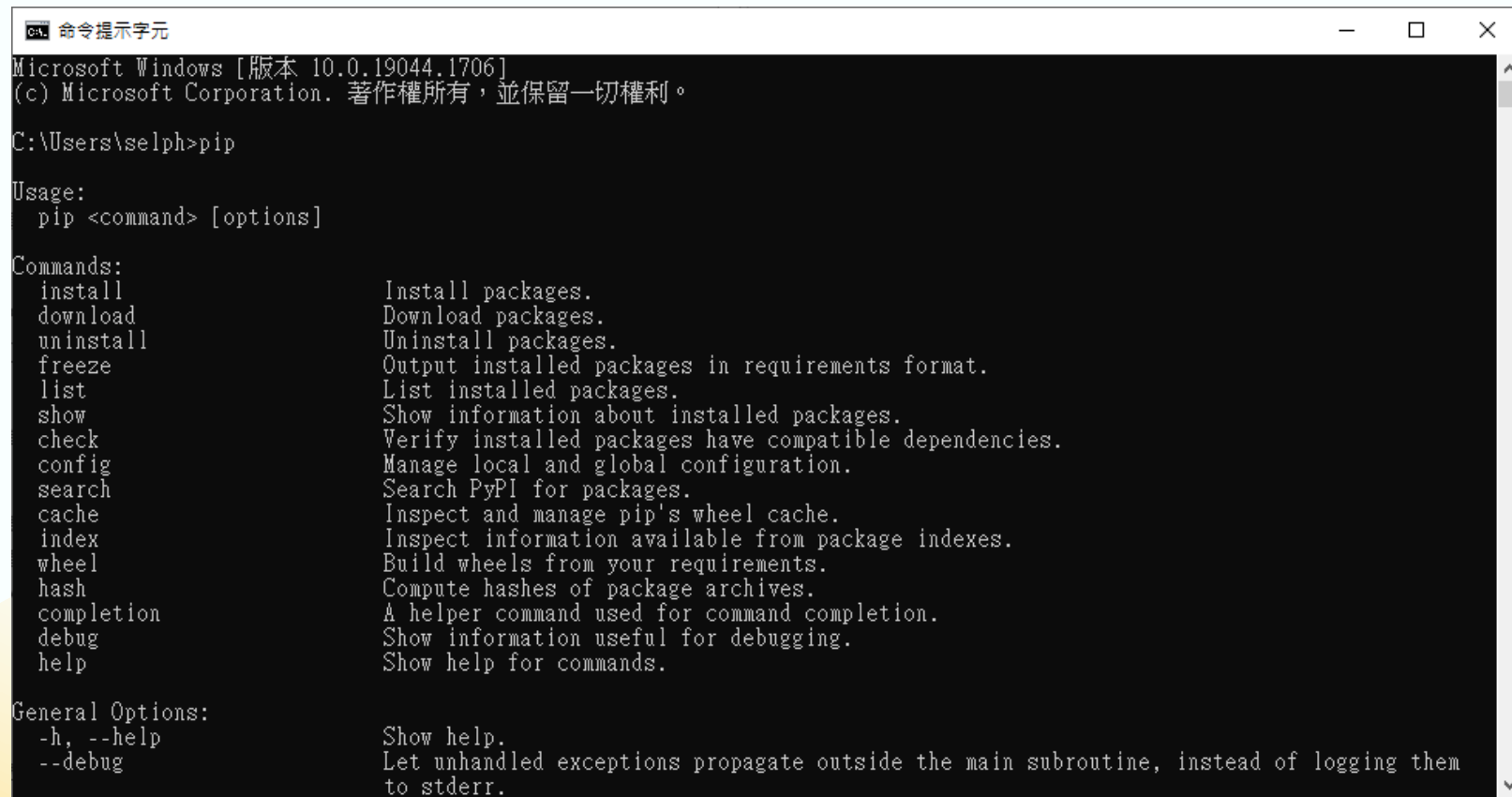
- ◆ Django需要使用官方python，請至python官網[下載](#)
- ◆ 安裝時請將下圖紅框處打勾



紅框處需打勾，否則就要
手動增加環境變數

Step 2-1. 確認python已安裝成功

- ◆ 打開命令提示字元，並輸入 `pip`，若有出現以下圖示中的文字則表示python有正常生效



```
命令提示字元
Microsoft Windows [版本 10.0.19044.1706]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\selp>pip

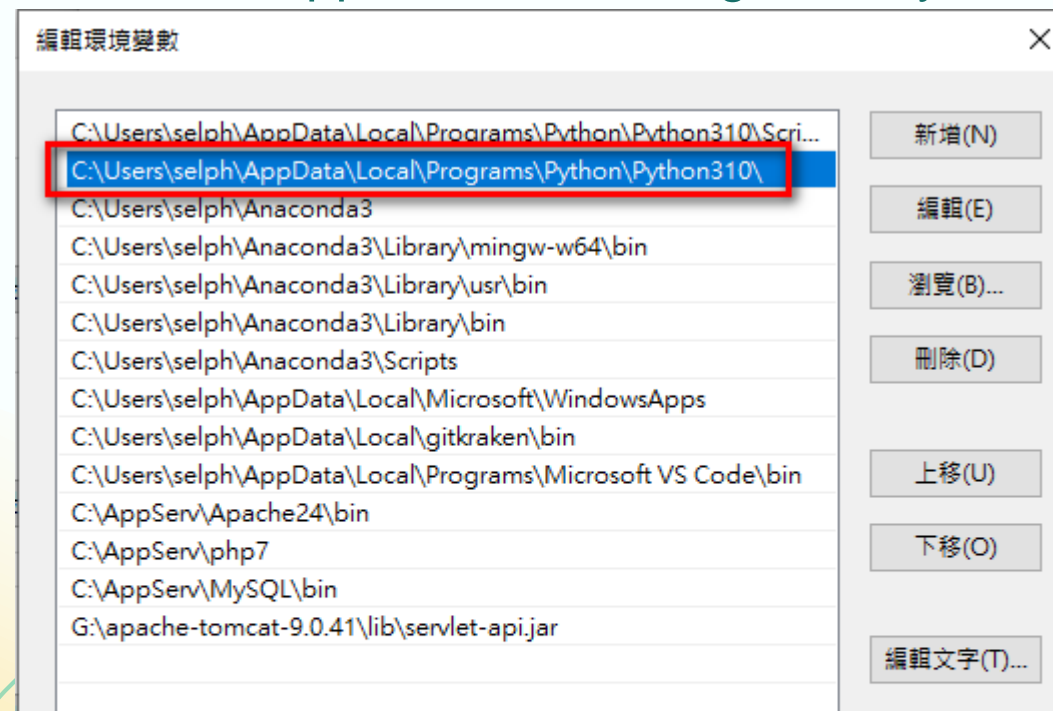
Usage:
  pip <command> [options]

Commands:
  install           Install packages.
  download          Download packages.
  uninstall         Uninstall packages.
  freeze            Output installed packages in requirements format.
  list              List installed packages.
  show              Show information about installed packages.
  check             Verify installed packages have compatible dependencies.
  config            Manage local and global configuration.
  search            Search PyPI for packages.
  cache             Inspect and manage pip's wheel cache.
  index             Inspect information available from package indexes.
  wheel             Build wheels from your requirements.
  hash              Compute hashes of package archives.
  completion        A helper command used for command completion.
  debug             Show information useful for debugging.
  help              Show help for commands.

General Options:
  -h, --help        Show help.
  --debug           Let unhandled exceptions propagate outside the main subroutine, instead of logging them to stderr.
```


Step 2-2. 確認python已安裝成功

- ◆ 若輸入pip出現「無法辨識‘pip’ 詞彙是否為 Cmdlet、函數、指令檔...」訊息時，請確認環境變數設定
- ◆ 我的電腦→內容→進階系統設定→進階→環境變數
- ◆ 系統預設python安裝路徑：
 - ◆ C:\Users\使用者名稱\AppData\Local\Programs\Python\Python310\



Setp 3. 安裝Django

- ◆ 輸入 `pip install django`
- ◆ 請用系統管理員身份執行

系統管理員: 命令提示字元

```
Microsoft Windows [版本 10.0.19044.2130]  
(c) Microsoft Corporation. 著作權所有，並保留一切權利。
```

```
C:\WINDOWS\system32>pip install django
```

```
Collecting django
```

```
  Downloading Django-4.1.2-py3-none-any.whl (8.1 MB)
```

```
----- 8.1/8.1 MB 7.4 MB/s eta 0:00:00
```

```
Requirement already satisfied: sqlparse>=0.2.2 in c:\users\selp\appdata\local\programs\python\python310\lib\site-packages (from django) (0.4.2)
```

```
Requirement already satisfied: asgiref<4,>=3.5.2 in c:\users\selp\appdata\local\programs\python\python310\lib\site-packages (from django) (3.5.2)
```

```
Requirement already satisfied: tzdata in c:\users\selp\appdata\local\programs\python\python310\lib\site-packages (from django) (2022.1)
```

```
Installing collected packages: django
```

```
Successfully installed django-4.1.2
```

開始django專案

Start Django

Step 1. 建立django專案

- ◆ 在電腦裡建立一個資料夾，並開啟命令提示字元切換至該資料夾下
- ◆ 範例在桌面建立名叫django的資料夾，開啟cmd後切換到該資料夾下

```
命令提示字元
Microsoft Windows [版本 10.0.19044.2130]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。
C:\Users\selp>cd C:\Users\selp\Desktop\django
C:\Users\selp\Desktop\django>
```

Step 2. 建立django專案

- ◆ 在命令列裡輸入以下指令：`django-admin startproject` 專案名

```
cmd 命令提示字元
Microsoft Windows [版本 10.0.19044.2130]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。
C:\Users\selp\ph>cd C:\Users\selp\ph\Desktop\django
C:\Users\selp\ph\Desktop\django>django-admin startproject mysite
C:\Users\selp\ph\Desktop\django>_
```

- ◆ 接著請到django資料夾下可以看到以一個叫mysite的子目錄，結構如下圖

```
▼ DJANGO
  ▼ mysite
    ▼ mysite
      __init__.py
      asgi.py
      settings.py
      urls.py
      wsgi.py
      manage.py
```

專案內容說明

檔案	說明
manage.py	Python命令程式，提供使用者方便管理專案的功能，包含建置專案、啟動server等功能
mysite	該專案的主目錄，包含重要的設定檔和RootURLConf(根URL配置檔)
__init__.py	一個空檔案，讓該目錄成為一個python package
setting.py	本專案的設定檔
urls.py	URL的根配置檔
wsgi.py	網頁伺服器與django應用介接檔案，可以視為django的進入點

Step 3-1. 首次啟動專案

- ◆ 切換到mysite資料夾下後輸入指令：`python manage.py runserver`
- ◆ 若能正常啟動將會看到server在幾點啟動、網址為何...等資訊
- ◆ 若要停止server，按下CTRL+C即可

```
C:\Users\selp\h\Desktop\django>cd mysite
```

```
C:\Users\selp\h\Desktop\django\mysite>python manage.py runserver
```

```
Watching for file changes with StatReloader
```

```
Performing system checks...
```

```
System check identified no issues (0 silenced).
```

```
You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
```

```
Run 'python manage.py migrate' to apply them.
```

```
October 31, 2022 - 20:13:01
```

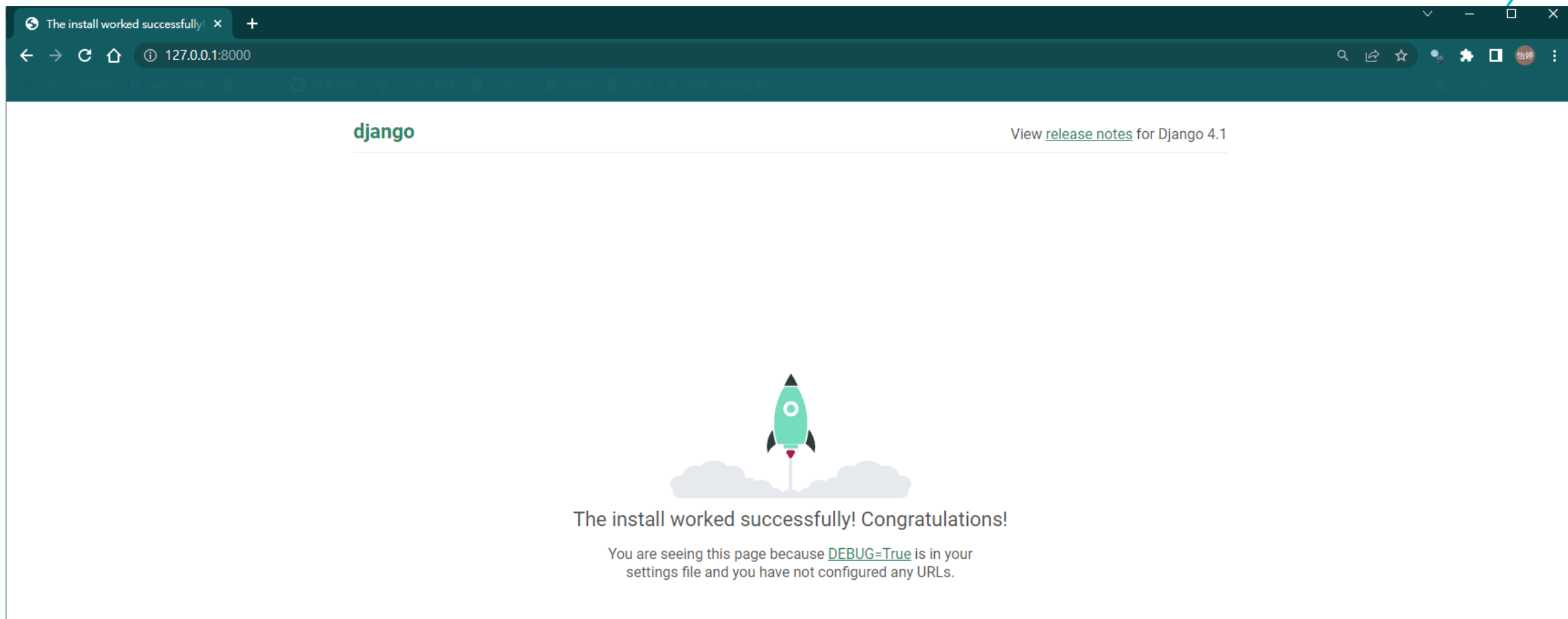
```
Django version 4.1.2, using settings 'mysite.settings'
```

```
Starting development server at http://127.0.0.1:8000/
```

```
Quit the server with CTRL-BREAK.
```

Step 3-2. 首次啟動專案

- ◆ 接著開啟網頁輸入剛django給你的URL，你就會看到django運作的畫面



Step 4. 調整基本設定 – 調整語系與時區

- ◆ 到mysite\mysite\setting.py
- ◆ 將LANGUAGE_CODE修改為zh-Hant(繁中)
- ◆ 將TIME_ZONE修改為Asia/Taipei

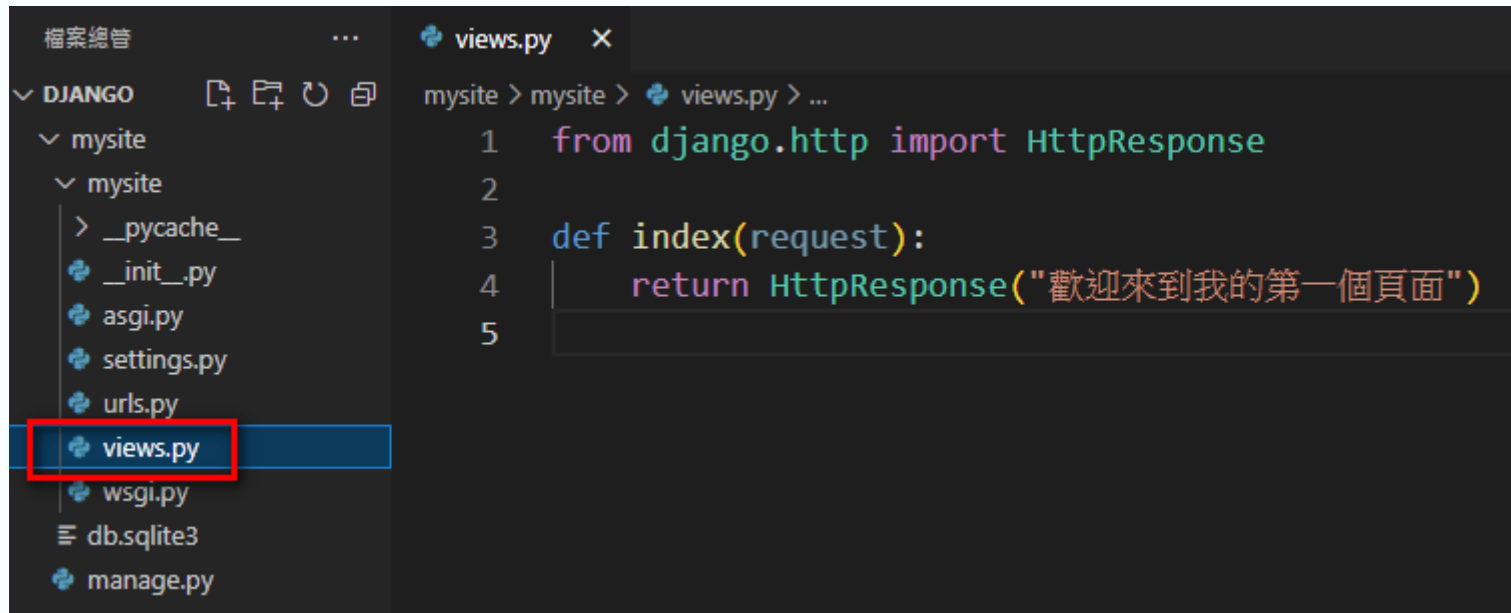
```
LANGUAGE_CODE = 'en-us'  
  
TIME_ZONE = 'UTC'  
  
USE_I18N = True  
  
USE_TZ = True
```



```
LANGUAGE_CODE = 'zh-Hant'  
  
TIME_ZONE = 'Asia/Taipei'  
  
USE_I18N = True  
  
USE_TZ = True
```

製作第一個頁面 Step 1

- ◆ 在mysite\mysite下新增views.py，並新增以下程式碼

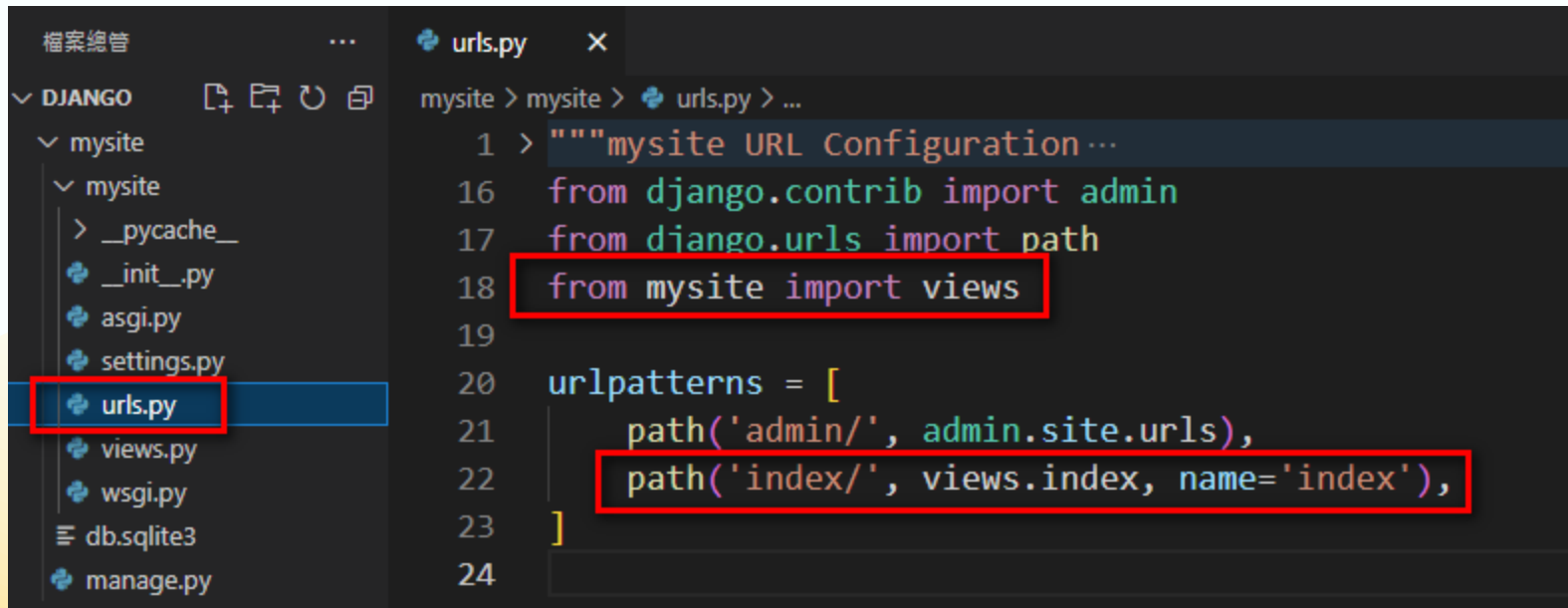


The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows the project structure: DJANGO > mysite > mysite. The file views.py is highlighted with a red box. The code editor shows the contents of views.py:

```
mysite > mysite > views.py > ...  
1  from django.http import HttpResponse  
2  
3  def index(request):  
4      return HttpResponse("歡迎來到我的第一個頁面")  
5
```

製作第一個頁面 Step 2

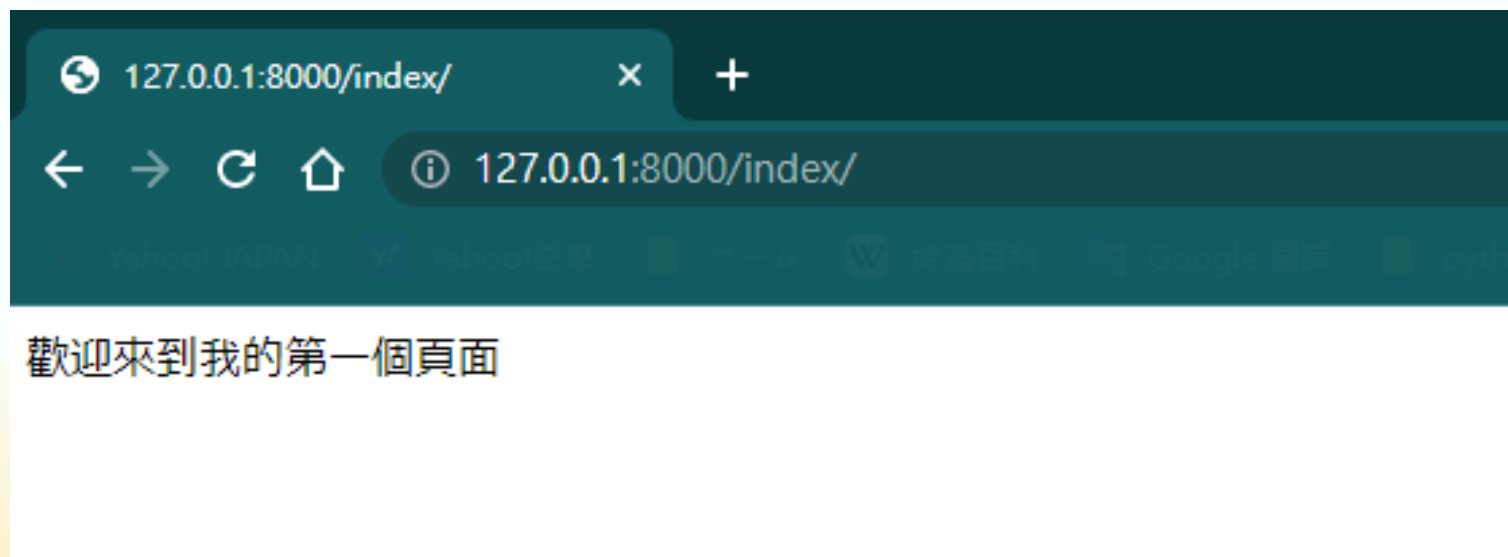
- ◆ 修改mysite\mysite\urls.py，添加紅框的部份
- ◆ from mysite import views
 - ◆ 從mysite裡匯入views.py這隻程式
- ◆ path('index/', views.index, name='index'),
- ◆ path(URL, funvtion或class, name=URL名稱)



```
mysite > mysite > urls.py > ...
1 > """mysite URL Configuration...
16 from django.contrib import admin
17 from django.urls import path
18 from mysite import views
19
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('index/', views.index, name='index'),
23 ]
24
```

製作第一個頁面 Step 3

- ◆ 在瀏覽器輸入`http://127.0.0.1:8000/index` 就可以看到剛做好的頁面了
- ◆ 若已有自訂頁面後，`http://127.0.0.1:8000`的火箭首頁會失效，此為正常狀況



建立APP專案

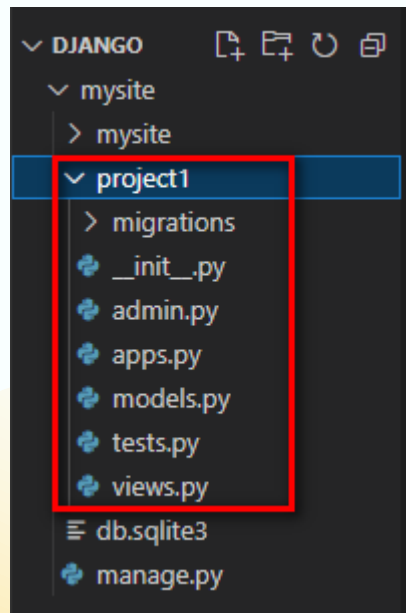
Create APP project

Step 1. 建立APP專案

- ◆ 到mysite資料夾下輸入指令 `py manage.py startapp 專案名`

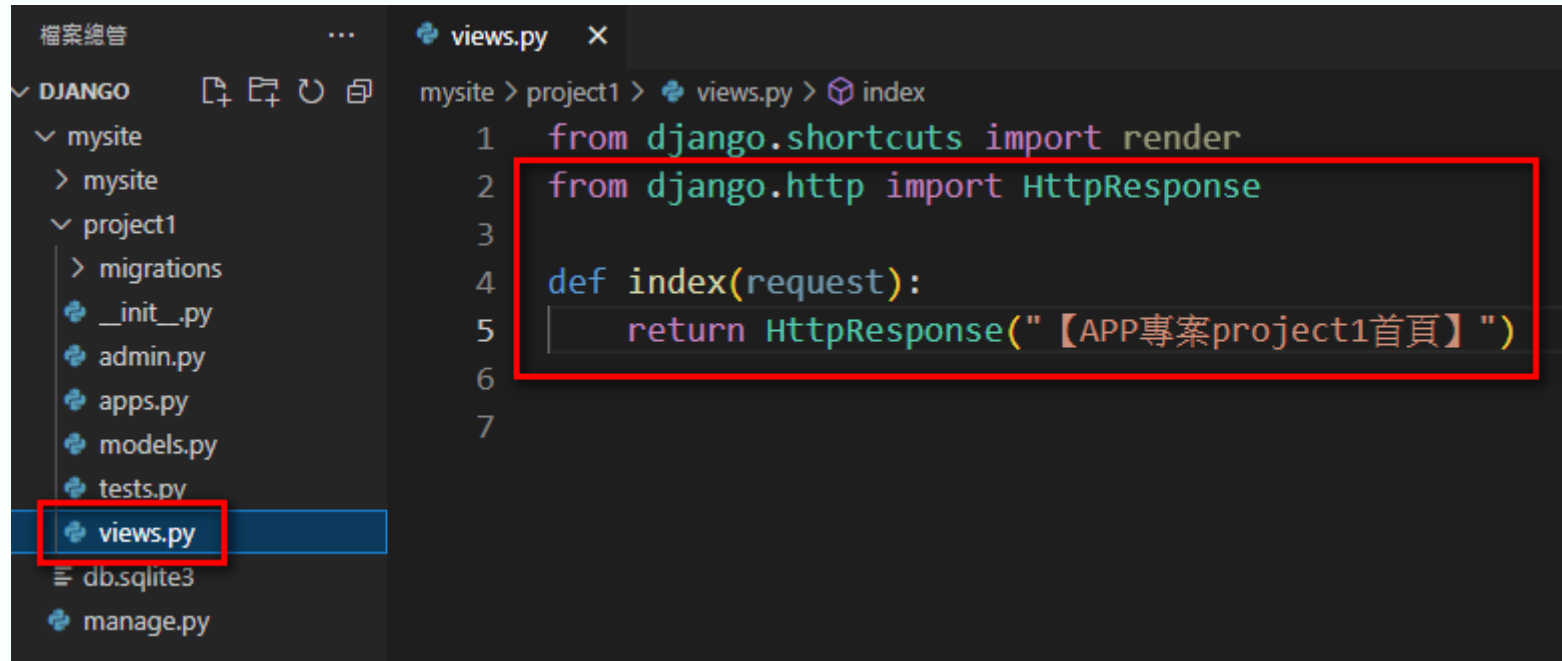
```
C:\Users\selp\ Desktop \django \mysite > py manage.py startapp project1  
C:\Users\selp\ Desktop \django \mysite > .
```

- ◆ 接著到project1資料夾下可以看到以下結構(紅框處)



Step 2. 修改views.py

- ◆ 修改mysite\project1\views.py，添加紅框的部份

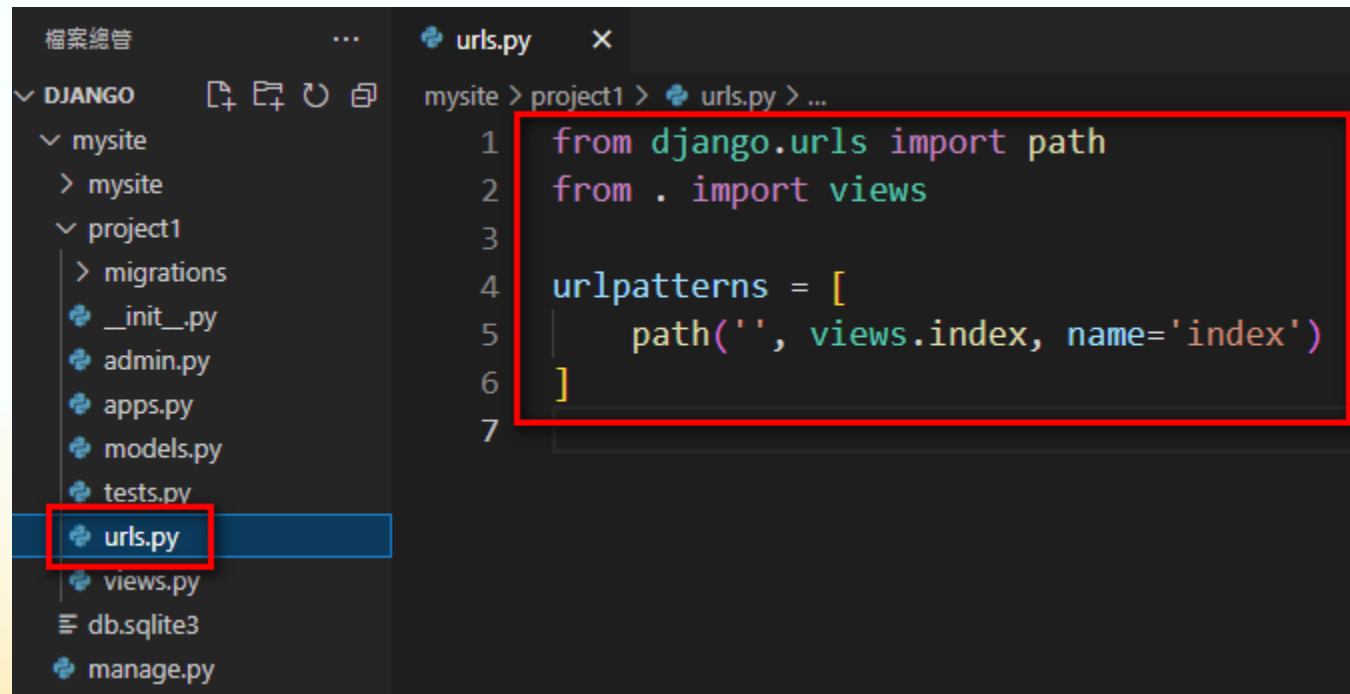


```
檔案總管
└─ DJANGO
   └─ mysite
      └─ mysite
         └─ project1
            ├── migrations
            ├── __init__.py
            ├── admin.py
            ├── apps.py
            ├── models.py
            ├── tests.py
            └─ views.py
               └─ db.sqlite3
                  └─ manage.py

views.py
mysite > project1 > views.py > index
1  from django.shortcuts import render
2  from django.http import HttpResponseRedirect
3
4  def index(request):
5      return HttpResponseRedirect("【APP專案project1首頁】")
6
7
```

Step 3. 新增urls.py

- ◆ 在mysite\project1下新增urls.py，並添加以下程式碼
- ◆ `path("", views.index, name='index')`
- ◆ `path(URL, views.function name, name=URL名稱)`

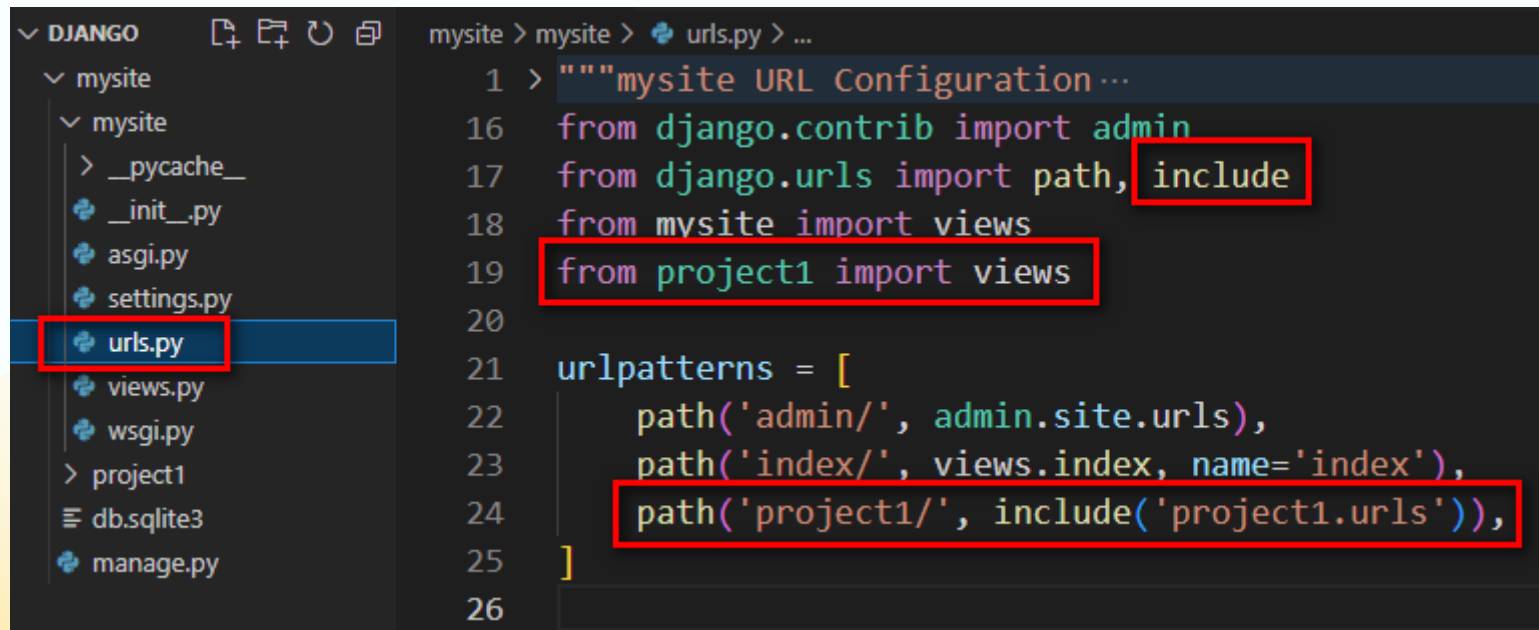


The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows the project structure: DJANGO > mysite > project1. The file `urls.py` is selected and highlighted with a red box. The code editor shows the following code:

```
1 from django.urls import path
2 from . import views
3
4 urlpatterns = [
5     path('', views.index, name='index')
6 ]
7
```


Step 4. 修改mysite\urls.py

- ◆ 在from django.urls import path後增加 include
- ◆ 匯入 from project1 import views
- ◆ urlpatterns新增path('members/', include('members.urls')),
- ◆ path(URL, include(APP專案名.urls))



```
mysite > mysite > urls.py > ...  
1 > """mysite URL Configuration...  
16 from django.contrib import admin  
17 from django.urls import path, include  
18 from mysite import views  
19 from project1 import views  
20  
21 urlpatterns = [  
22     path('admin/', admin.site.urls),  
23     path('index/', views.index, name='index'),  
24     path('project1/', include('project1.urls')),  
25 ]  
26
```

Step 5. 啟動server

- ◆ 在命令列執行 `py manage.py runserver`
- ◆ 到瀏覽器裡輸入 <http://127.0.0.1:8000/project1/>



執行示意圖

mysite\urls.py(廣域的設定)

```
mysite > mysite > urls.py > ...
1 > """mysite URL Configuration...
16 from django.contrib import admin
17 from django.urls import path, include
18 from mysite import views
19 from project1 import views
20
21 urlpatterns = [
22     path('admin/', admin.site.urls),
23     path('index/', views.index, name='index'),
24     path('project1/', include('project1.urls')),
25 ]
```

project1\urls.py(區域的設定)

```
mysite > project1 > urls.py > ...
1 from django.urls import path
2 from . import views
3
4 urlpatterns = [
5     path('', views.index, name='index'),
6 ]
```

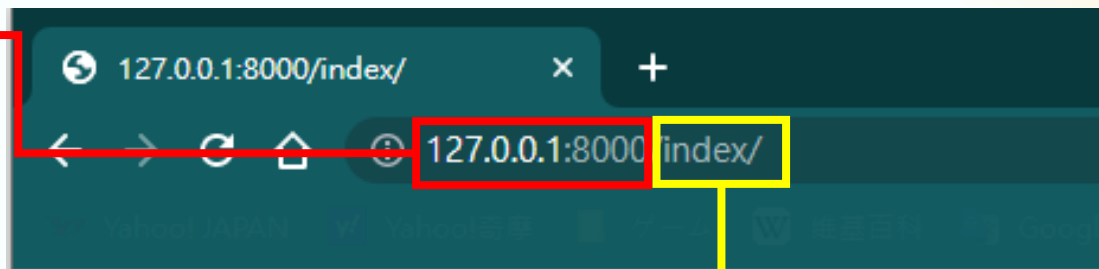
project1\views.py

```
mysite > project1 > views.py > ...
1 from django.shortcuts import render
2 from django.http import HttpResponse
3
4 def index(request):
5     return HttpResponse("【APP專案-project1首頁】")
```

使用include後，被include的urls都會自動加上project1

【APP專案-project1首頁】

【APP專案-project1首頁】



後台管理系統

Admin

創立新專案

- ◆ 在命令列輸入 `py manage.py startapp 專案名`

```
C:\Users\selp\h\Desktop\django\mysite>py manage.py startapp restaurants  
C:\Users\selp\h\Desktop\django\mysite>
```

- ◆ 我們要對這個專案建立資料庫，之後這專案的資料都會放在該資料庫下，在命令列輸入 `py manage.py migrate`

```
C:\Users\selp\h\Desktop\django\mysite>py manage.py migrate  
Operations to perform:  
  Apply all migrations: admin, auth, contenttypes, sessions  
Running migrations:  
  Applying contenttypes.0001_initial... OK  
  Applying auth.0001_initial... OK  
  Applying admin.0001_initial... OK  
  Applying admin.0002_logentry_remove_auto_add... OK  
  Applying admin.0003_logentry_add_action_flag_choices... OK  
  Applying contenttypes.0002_remove_content_type_name... OK  
  Applying auth.0002_alter_permission_name_max_length... OK  
  Applying auth.0003_alter_user_email_max_length... OK  
  Applying auth.0004_alter_user_username_opts... OK  
  Applying auth.0005_alter_user_last_login_null... OK  
  Applying auth.0006_require_contenttypes_0002... OK  
  Applying auth.0007_alter_validators_add_error_messages... OK  
  Applying auth.0008_alter_user_username_max_length... OK  
  Applying auth.0009_alter_user_last_name_max_length... OK  
  Applying auth.0010_alter_group_name_max_length... OK  
  Applying auth.0011_update_proxy_permissions... OK  
  Applying auth.0012_alter_user_first_name_max_length... OK  
  Applying sessions.0001_initial... OK  
C:\Users\selp\h\Desktop\django\mysite>_
```

創立超級管理員

- ◆ django有一個後台系統，但初始是沒有管理員存在的，所以必需手動設置，在命令列輸入 `py manage.py createsuperuser`
- ◆ 輸入使用者名稱
- ◆ 輸入電子信箱
- ◆ 輸入密碼
- ◆ 再次確認密碼
- ◆ 確認是否要創建使用者

```
C:\Users\selph\Desktop\django\mysite>py manage.py createsuperuser
使用者名稱 (leave blank to use 'selph'): root
電子信箱: selphie20150501@gmail.com
Password:
Password (again):
這個密碼太普通。
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
C:\Users\selph\Desktop\django\mysite>
```

進入後台管理系統

◆ 到後台管理系統 <http://127.0.0.1:8000/admin/> (如下圖)

◆ 登入成功後會看到下圖



The screenshot shows the Django Admin interface. At the top is a dark blue header with the text "Django 管理" in yellow. Below the header is a white section titled "網站管理". Under this, there is a blue bar labeled "認證與授權". Below this bar, there are two rows: "使用者" and "群組". Each row has a green plus icon and the text "新增" followed by a yellow pencil icon and the text "變更".

認證與授權	
使用者	+ 新增 變更
群組	+ 新增 變更



The screenshot shows the Django Admin login form. It has a dark blue header with the text "Django 管理". Below the header, there are two input fields: "使用者名稱:" and "密碼:". Below the "密碼:" field is a blue button labeled "登入".

Django 管理	
使用者名稱:	<input type="text"/>
密碼:	<input type="password"/>
<input type="button" value="登入"/>	

建立模型

- ◆ 修改mysite\restaurants\models.py
- ◆ 此意思是指restaurants\models.py宣告一個Restaurant類別，並定義裡面的屬性，django會依據這個建立資料表以及欄位設定

```
mysite > restaurants > models.py > ...  
1  from django.db import models  
2  
3  class Restaurant(models.Model):  
4      name = models.CharField(max_length=20)  
5      phone_number = models.CharField(max_length=15)  
6      address = models.CharField(max_length=50)
```

有關Field的說明可以參考[django官網](#)

追蹤APP

- ◆ 當專案設置完成時，通常都需要請django來追蹤
- ◆ 到mysite\settings.py裡找到INSTALLED_APPS後，在裡面添加你剛設置完成的APP專案名

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'restaurants',  
]
```

建立migration資料檔

- ◆ 當模型建立完成後，首先要先檢查剛建立的是否正確，在命令列輸入 `py manage.py check`
- ◆ 若資料都正確無誤，系統會顯示System check identified no issues (0 silenced).
- ◆ 資料都正確無誤後就可以建立migrations資料檔，確保資料表架構與版本都有被記錄，在命令列輸入 `py manage.py makemigrations 專案名`


```
C:\Users\selp\h\Desktop\django\mysite>py manage.py check
System check identified no issues (0 silenced).
```

```
C:\Users\selp\h\Desktop\django\mysite>py manage.py makemigrations restaurants
Migrations for 'restaurants':
  restaurants\migrations\0001_initial.py
  - Create model Restaurants
C:\Users\selp\h\Desktop\django\mysite>
```

更新migrations資料檔

- ◆ 若我們對models.py裡做修改，把地址的長度從50改40
- ◆ 一樣先確認資料是否正確，接著再建立migrations資料檔，此時就可以看到系統產出0002號的migrations記錄

```
mysite > restaurants > models.py > ...  
1  from django.db import models  
2  
3  class Restaurant(models.Model):  
4      name = models.CharField(max_length=20)  
5      phone_number = models.CharField(max_length=15)  
6      address = models.CharField(max_length=40)  
7
```



```
C:\Users\selp\h\Desktop\django\mysite>py manage.py check  
System check identified no issues (0 silenced).  
  
C:\Users\selp\h\Desktop\django\mysite>py manage.py makemigrations restaurants  
Migrations for 'restaurants':  
  restaurants\migrations\0002_alter_restaurants_address.py  
    - Alter field address on restaurants  
  
C:\Users\selp\h\Desktop\django\mysite>
```

模型與資料庫同步

- ◆ 前面剛剛都只是先建立migrations記錄而已，都還沒跟資料庫進行同步，在與資料庫同步前必需再三確認你的資料表都沒問題才同步
- ◆ 同步時請輸入 `py manage.py migrate` 專案名 版號

```
C:\Users\selp\ Desktop \django \mysite > py manage.py migrate restaurants 0001
Operations to perform:
  Target specific migration: 0001_initial, from restaurants
Running migrations:
  Applying restaurants.0001_initial... OK
C:\Users\selp\ Desktop \django \mysite >
```

讓django後台追蹤資料庫

- ◆ 剛我們已經讓模型與資料庫同步了，但登入後台卻沒看到資料庫，原因為需手動設置讓後台顯示資料庫
- ◆ 到mysite\restaurants\admin.py裡，增加紅框處程式碼

```
mysite > restaurants > admin.py
1  from django.contrib import admin
2  from .models import Restaurant
3
4  admin.site.register(Restaurant)
```

讓django後台追蹤資料庫

- ◆ 重整後台後就可以看到增加了「restaurants」的資料庫，而且也可以手動一筆一筆新增資料

Django 管理

網站管理

RESTAURANTS

Restaurantss

+ 新增

✎ 變更

最近的動作

我的動作

認證與授權

使用者

+ 新增

群組

+ 新增

新增 restaurants

Name :

Phone number :

Address :

資料操作

- ◆ 資料可以透過shell功能手動一筆一筆增加
- ◆ 在命令列裡輸入`py manage.py shell`，進入shell後就跟一般python一樣輸入指令
- ◆ 新增完畢後可以到django後台去確認資料是否已新增

```
C:\Users\selp\h\Desktop\django\mysite>py manage.py shell
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.5.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: from restaurants.models import Restaurant

In [2]: r1 = Restaurant(name="巷口小吃店", phone_number="02-123456", address="台北市中山北路一段50號")

In [3]: r1.save()

In [4]: r2 = Restaurant(name="我家鍋貼", phone_number="02-333546", address="台北市信義路三段180號")

In [5]: r2.save()
```

遠端連線

下載ngrok

- ◆ 到官網下載ngrok <https://ngrok.com/>
- ◆ 該網站需要先登入後才能下載

Download ngrok

ngrok is easy to install. Download a single binary with zero run-time dependencies.

📄 Download for Windows



Mac OS
Mac OS (ARM64)



Windows
Windows (32-Bit)



Linux
Linux (32-Bit)
Linux (ARM)
Linux (ARM64)



FreeBSD
FreeBSD (32-Bit)

取得ngrok的token

- ◆ 當你登入ngrok官網後請看到步驟2的「Connect your account」，裡面有你的token(每個人都不一樣)

2. Connect your account

Running this command will add your authtoken to the default `ngrok.yml` configuration file. This will grant you access to more features and longer session times. Running tunnels will be listed on the [endpoints page](#) of the dashboard.

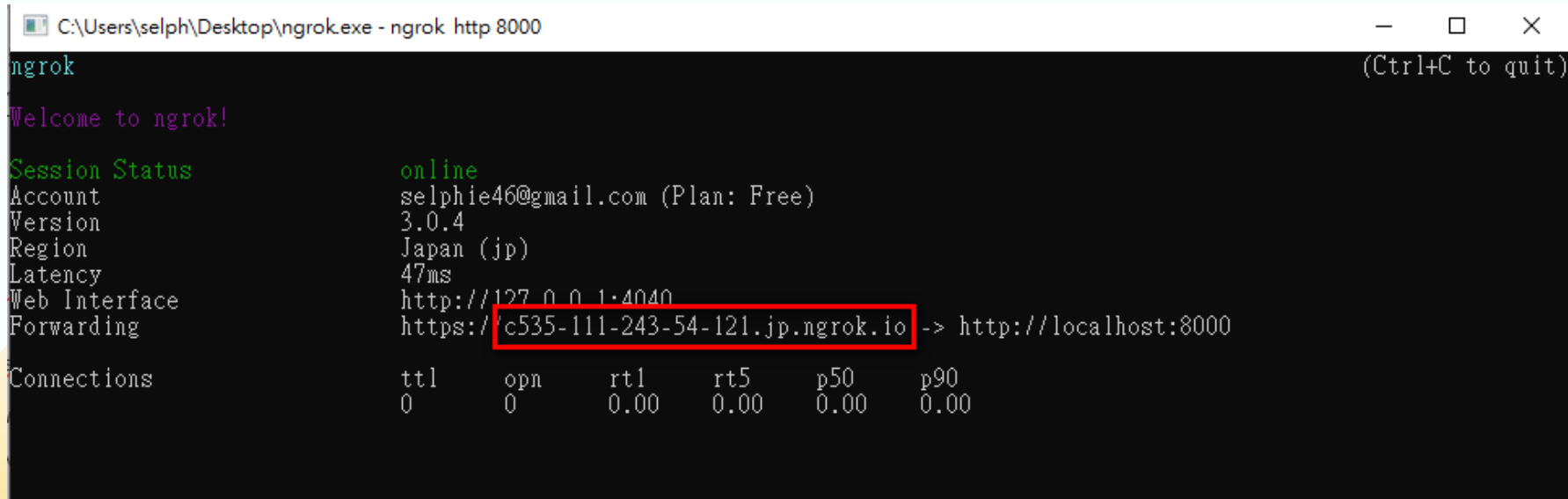
```
$ ngrok config add-authtoken 2Gz1h4WTC51dnoSvk9L9TV3WK0n_6rsLMJp5m7hF8YVcnq8GT
```

- ◆ 執行剛解壓縮的ngrok.exe，並輸入官網給你的token

```
ngrok is a command line application, try typing 'ngrok.exe http 80'
at this terminal prompt to expose port 80.
C:\Users\selp\h\Desktop>ngrok config add-authtoken 2Gz1h4WTC51dnoSvk9L9TV3WK0n_6rsLMJp5m7hF8YVcnq8GT
Authtoken saved to configuration file: C:\Users\selp\h\AppData\Local\ngrok\ngrok.yml
C:\Users\selp\h\Desktop>
```

啟動ngrok

- ◆ 輸入 **ngrok http 8000**
- ◆ 啟動後會看到以下畫面
- ◆ 若啟動時會閃退，或出現「Your account is limited to 1 simultaneous ngrok agent session.」訊息，請輸入 **tskill ngrok**
- ◆ 啟動ngrok前請先把django服務停止



A screenshot of a Windows command prompt window titled "C:\Users\selp\\Desktop\ngrok.exe - ngrok http 8000". The window shows the output of the ngrok command. It starts with "ngrok" and "(Ctrl+C to quit)". Then it says "Welcome to ngrok!". Below that is a "Session Status" section. The status is "online". The account is "selphie46@gmail.com (Plan: Free)". The version is "3.0.4". The region is "Japan (jp)". The latency is "47ms". The web interface is "http://127.0.0.1:4040". The forwarding is "https://c535-111-243-54-121.jp.ngrok.io -> http://localhost:8000". At the bottom, there is a table for "Connections" with columns: "ttl", "opn", "rt1", "rt5", "p50", and "p90". The values are: "0", "0", "0.00", "0.00", "0.00", and "0.00".

```
C:\Users\selp\Desktop\ngrok.exe - ngrok http 8000
ngrok
Welcome to ngrok!

Session Status
Account      selphie46@gmail.com (Plan: Free)
Version      3.0.4
Region      Japan (jp)
Latency      47ms
Web Interface http://127.0.0.1:4040
Forwarding   https://c535-111-243-54-121.jp.ngrok.io -> http://localhost:8000

Connections
ttl      opn      rt1      rt5      p50      p90
0         0         0.00     0.00     0.00     0.00
```

■ 修改settings.py

- ◆ 到mysite\mysite\settings.py找到ALLOWED_HOSTS
- ◆ 將上一步驟的紅框處添加至ALLOWED_HOSTS裡
- ◆ 每次重啟ngrok服務時，系統都會重給一組連線位置，拿到新位置後需再次修改settings.py的host位置

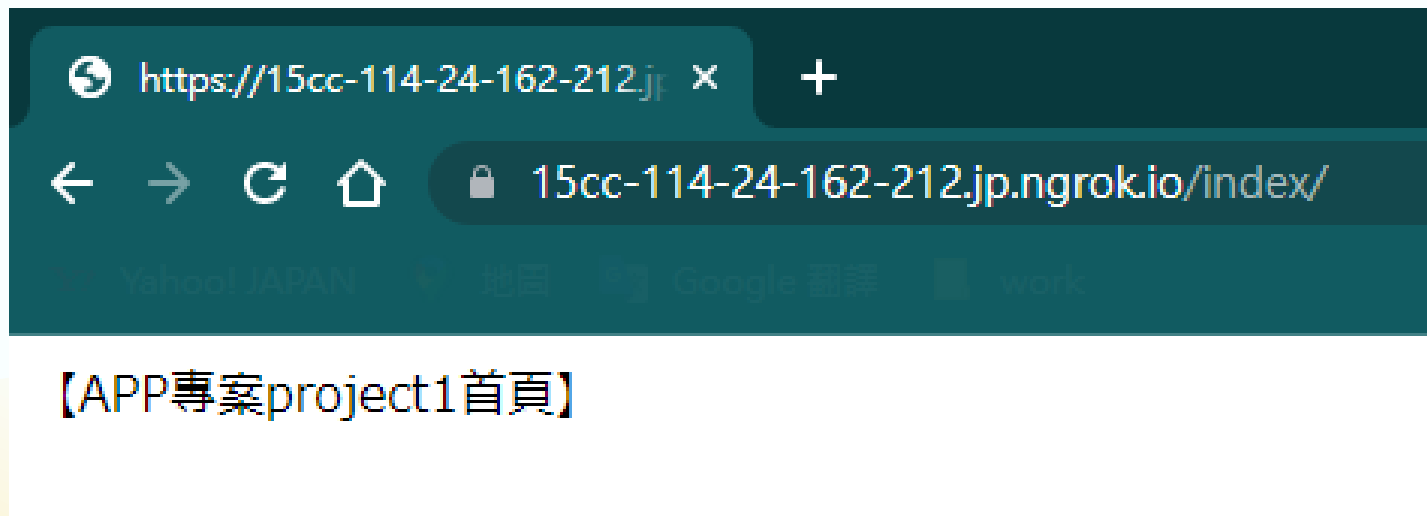
```
# SECURITY WARNING: don't run with debug turned on in p
DEBUG = True

ALLOWED_HOSTS = ['c535-111-243-54-121.jp.ngrok.io']
#ALLOWED_HOSTS = ['*']

# Application definition
```

開啟瀏覽器

- ◆ 在瀏覽器裡輸入ngrok的Forwarding後那串網址+專案名稱，其他使用者就可以瀏覽你的django頁面了
- ◆ 範例<https://15cc-114-24-162-212.jp.ngrok.io/index/>



遠端連線

分享IP

無線分享器設定

- ◆ 此範例使用ASUS分享器，至分享器設定頁面→外部網路(WAN)→虛擬伺服器→新增設定檔
- ◆ ip可至cmd輸入ipconfig查詢，看IPv4 位址那組

The screenshot displays the ASUS RT-AC52 web management interface. The left sidebar shows the navigation menu with '外部網路(WAN)' (External Network) highlighted. The main content area is titled '外部網路(WAN) - 虛擬伺服器' (External Network (WAN) - Virtual Server). It includes a description of virtual servers and a table for port forwarding rules.

ASUS RT-AC52 登入 重新開機 繁體中文

運作模式: 無線路由器 韌體版本: 3.0.0.4.382_70222 SSID:

網際網路連線 通訊埠觸發程式 虛擬伺服器 DMZ DDNS NAT Passthrough

外部網路(WAN) - 虛擬伺服器

虛擬伺服器 / 通訊埠轉發 (Port Forwarding) 可允許在 Internet 的設備連到架設在區域網路 (LAN) 內的伺服器。欲加速連線，一些 P2P 應用程式 (如 BitTorrent) 也需要您設定通訊埠轉發。更多資訊請參考 P2P 應用程式使用手冊。

若您要為同一個網路中的用戶端指定埠範圍，請輸入服務名稱、埠範圍 (10200:10300)、網路 IP 位址，但不要填寫本地埠。

- 在關閉那防火牆時設定 HTTP 伺服器，且設定外網 (WAN 端) 通訊埠為 80，會與 RT-AC52 的管理介面伺服器使用的通訊埠衝突。

[虛擬伺服器 FAQ](#)

基本設定

開啟虛擬伺服器 ☒

通訊埠轉發清單 (最多限制: 64)

服務名稱	外部通訊埠	內部通訊埠	本地 IP 位址	通訊協定	來源 IP	編輯	刪除
				TCP			
				BOTH			
				BOTH			
django	8000	8000	192.168.1.5	BOTH			

新增設定檔

對外IP

- ◆ 回到無線分享器首頁，可以看到電信公司給你的IP
- ◆ 或上網查詢也可以 <https://nordvpn.com/zh-tw/ip-lookup/>

IP 位址查詢

請輸入您想知道的 IP 位址：

取得 IP 詳細資料

 以您的 IP 為例

修改settings.py

- ◆ 到testsite\testsite\settings.py找到ALLOWED_HOSTS
- ◆ 修改為ALLOWED_HOSTS = ['*'] (註：修改成*會讓所有IP都通過，有風險)

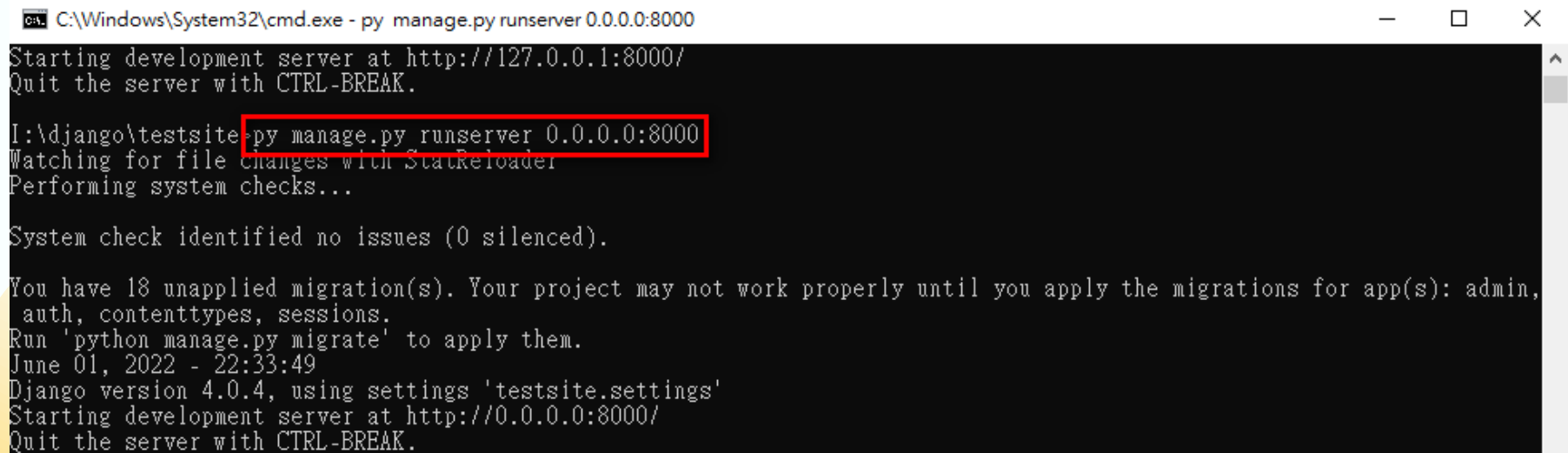
```
DEBUG = True

#ALLOWED_HOSTS = ['c535-111-243-54-121.jp.ngrok.io']
ALLOWED_HOSTS = ['*']

# Application definition
```

啟動 Django

- ◆ 輸入 `py manage.py runserver 0.0.0.0:8000`
- ◆ `runserver 0.0.0.0:0` 則指 socket 會監聽該電腦所有網卡的 ip。例如你電腦中有以下三組 ip，系統會在這三組 ip 中隨機找一組跟 socket 連接
 - ◆ 127.0.0.1 (本地端)
 - ◆ 192.168.1.xx (本地端)
 - ◆ 111.243.xx.xx (電信公司給你的)



```
C:\Windows\System32\cmd.exe - py manage.py runserver 0.0.0.0:8000
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

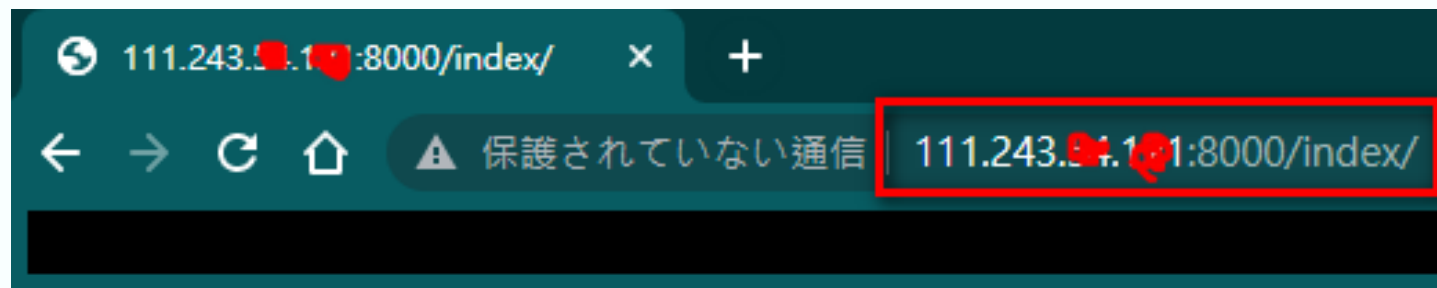
I:\django\testsite>py manage.py runserver 0.0.0.0:8000
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
June 01, 2022 - 22:33:49
Django version 4.0.4, using settings 'testsite.settings'
Starting development server at http://0.0.0.0:8000/
Quit the server with CTRL-BREAK.
```

開啟瀏覽器

- ◆ 在瀏覽器輸入你的ip，其他使用者就可以瀏覽你的django頁面
- ◆ 範例 `http://111.243.xx.xx:8000/index/`



いらっしゃいませ!!!