

➤ Django 本機端設定

安裝套件

pip install django

1. 建立 django 專案

django-admin startproject 專案名稱(例如 myDjango)

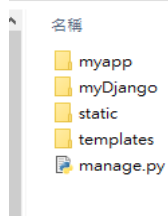
2. 建立 application 應用程式

cd 專案名稱

python manage.py startapp 應用程式名稱(例如 myapp)

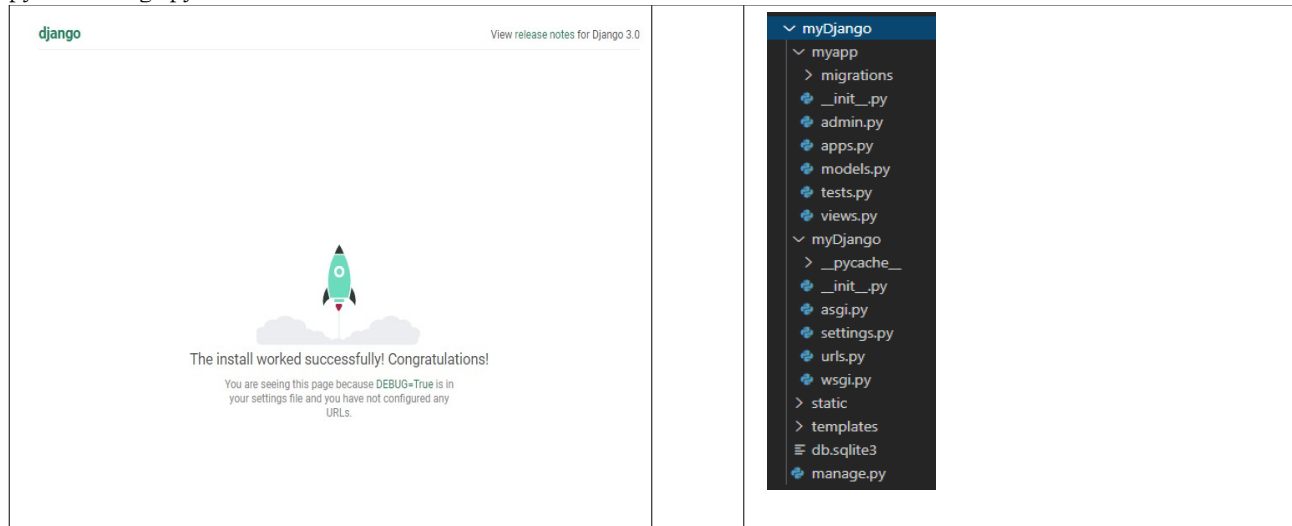
3. 建立 templates 和 static 資料夾

與專案名稱和應用程式同層，增加 templates 和 static 資料夾



4. 啟動 server

python manage.py runserver



5. 加入 application 應用程式

開啟 settings.py 檔案

INSTALLED_APPS = [

```
'django.contrib.admin',  
'django.contrib.auth',  
'django.contrib.contenttypes',  
'django.contrib.sessions',  
'django.contrib.messages',  
'django.contrib.staticfiles',  
'myapp' -->增加應用程式名稱
```

]

6. 設定 Template 路徑

開啟 settings.py 檔案

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')],
        'APP_DIRS': True,
        ........略
```

7. 設定語系和時區

開啟 settings.py 檔案

```
LANGUAGE_CODE = 'zh-Hant'
TIME_ZONE = 'Asia/Taipei'
```

8. 設定 urls.py

```
from django.contrib import admin
from django.urls import path
from django.conf.urls import url
from myapp.views import hello,hello01,hello02
```

```
urlpatterns = [
    path('admin/', admin.site.urls),
    url(r'^$', hello),
    url(r'^hello01/(\w+)/$', hello01),
    url(r'^hello02/(\w+)/$', hello02)
]
```

9. 定義函式

開啟 views.py 檔案

```
from django.shortcuts import render
from django.http import HttpResponse
from datetime import datetime
# Create your views here.
```

```
def hello(request):
    return HttpResponse("HELLO DJANGO!!")

def hello01(request, username):
    return HttpResponse('Hello~:'+username)

def hello02(request, username):
    now=datetime.now()
    return render(request, "hello02.html",locals())
```

10. 設定 template/hello02.html 檔

```
<!DOCTYPE html>
<html>
<head>
<title>第一個模版</title>
</head><body>
<h1>歡迎光臨~:{{ username }}</h1>
<h2>現在時刻~:{{ now }}</h2>
</body></html>
```

➤ Django 上傳 Heroku 遠端設定

1. Procfile(專案名稱.wsgi)

```
web: gunicorn myDjango.wsgi
```

2. requirements.txt 檔案設定

```
line-bot-sdk  
flask  
django  
gunicorn  
dj-static
```

3. wsgi.py 檔案設定

```
import os  
from django.core.wsgi import get_wsgi_application  
from dj_static import Cling  
os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'myDjango.settings')  
application = Cling(get_wsgi_application())
```

4. settings.py 檔案設定

```
ALLOWED_HOSTS = ['*']
```

錯誤參考

```
$ heroku config:set DISABLE_COLLECTSTATIC=1
```

輸出為 heroku config:set DISABLE_COLLECTSTATIC=1

➤ Django 和 Line 設定

1. 專案中的 urls.py

```
from django.contrib import admin
from django.urls import path, include
from django.conf.urls import url
from myapp.views import hello, hello01, hello02, callback
from myapp import views

urlpatterns = [
    path('admin/', admin.site.urls),
    url(r'^$', hello),
    url(r'^hello01/(\w+)/$', hello01),
    url(r'^hello02/(\w+)/$', hello02),
    url('^callback', views.callback),
]
```

2. 應用程式中新增 urls.py

```
from django.urls import path
from . import views

urlpatterns = [
    path('callback/', views.callback, name='callback'),
]
```

3. 應用程式 views.py

```
from django.shortcuts import render
from datetime import datetime
from django.conf import settings
from django.http import HttpRequest, HttpResponse, HttpResponseBadRequest
from django.views.decorators.csrf import csrf_exempt

from linebot import LineBotApi, WebhookHandler, WebhookParser
from linebot.exceptions import InvalidSignatureError
from linebot.models import MessageEvent, TextMessage, TextSendMessage

line_bot_api = LineBotApi('YOUR_CHANNEL_ACCESS_TOKEN')
parser = WebhookParser('YOUR_CHANNEL_SECRET')

# Create your views here.

def hello(request):
    return HttpResponse("hello Django")

def hello01(request, username):
    return HttpResponse("hello~:" + username)

def hello02(request, username):
    now = datetime.now()
    return render(request, "hello02.html", locals())

@csrf_exempt
def callback(request):
```

```
if request.method == 'POST':
    signature = request.META['HTTP_X_LINE_SIGNATURE']
    body = request.body.decode('utf-8')

    try:
        events = parser.parse(body, signature)
    except InvalidSignatureError:
        return HttpResponseForbidden()
    except LineBotApiError:
        return HttpResponseBadRequest()

    for event in events:
        if isinstance(event, MessageEvent):
            line_bot_api.reply_message(event.reply_token, TextSendMessage(text=event.message.text))

    return HttpResponse()
else:
    return HttpResponseBadRequest()
```