

Network Programming

- 網路程式設計 -

使用者管理

授課教師：張珀銀 老師



前一單元內容回顧

07

使用者管理

7-1 Django auth 簡介

7-2 登入和登出

1

Django auth 簡介

7-1 Django auth 簡介

7.1.1 安裝 Django auth App

```
33 INSTALLED_APPS = [  
34     'django.contrib.admin',  
35     'django.contrib.auth', # 確認已加入  
36     'django.contrib.contenttypes',  
37     'django.contrib.sessions',  
38     'django.contrib.messages',  
39     'django.contrib.staticfiles',  
40     'loginapp', # 新增的 app  
41 ]  
42  
43 MIDDLEWARE = [  
44     'django.middleware.security.SecurityMiddleware',  
45     'django.contrib.sessions.middleware.SessionMiddleware',  
46     'django.middleware.common.CommonMiddleware',  
47     'django.middleware.csrf.CsrfViewMiddleware',  
48     'django.contrib.auth.middleware.AuthenticationMiddleware',  
         # 確認已加入  
49     'django.contrib.messages.middleware.MessageMiddleware',  
50     'django.middleware.clickjacking.XFrameOptionsMiddleware',  
51 ]
```

7.1.2 開啟login專案

登入 admin 後台管理

輸入「`http://127.0.0.1:8000/admin/`」後，輸入管理帳號、密碼進入管理介面。



在 User 資料表新增使用者

點選 **使用者** 項目的 **新增** 鈕，以管理介面在 User 資料表新增一筆資料。



7.1.3 讀取 Django auth 使用者

例如：以`objects.all()` 方法讀取User 資料表所有資料，讀取時記得以「`from django.contrib.auth.models import User`」匯入模組。

```
from django.contrib.auth.models import User
User.objects.all()
```

例如：以`objects.get()` 方法取得帳號為「test」這筆資料，`objects.get()` 查詢成功會傳回1 筆資料，失敗會產生錯誤，引發例外，因此通常會以`try...except` 捕捉錯誤。

```
from django.contrib.auth.models import User
try:
    user=User.objects.get(username="test")
except:
    user=None
```

7.1.4 HttpRequest.user 物件

例如：建立index() 自訂函式顯示<index.html> 首頁，並判斷使用者是否認證過，若認證過，以username 取得其名稱。

```
def index(request):  
    if request.user.is_authenticated:  
        name=request.user.username  
    return render(request, "index.html", locals())
```


User 物件常用的屬性：

屬性	說明
username	使用者帳號。
first_name	名字。
last_name	姓氏。
email	電子信箱。
password	密碼 (加密)。
is_staff	布林值，設定是否可登入 admin 後台。
is_active	布林值，設定該使用者是否有效 (可登入)。
is_superuser	布林值，設定這個使用者用是否擁有全部權限而不需要給他分配明確的權限。
last_login	上一次登錄的日期時間。
date_joined	使用者創建的日期時間。

User 物件常用的方法：

方法	說明
get_username()	取得使用者帳號。
is_anonymous()	是否是匿名用戶，如果 HttpRequest.user 是 User 物件則傳回 False，如果 HttpRequest.user 是 AnonymousUser 物件則傳回 True。
is_authenticated()	使用者是否認證過。如果 HttpRequest.user 是 User 物件則傳回 True，如果 HttpRequest.user 是 AnonymousUser 物件則傳回 False。
get_full_name()	取得名字 (first_name) 和姓氏 (last_name)，中間有一空格。
get_short_name()	取得名字 (first_name)。

範例：新增使用者

本範例利用程式直接在auth的User資料表中新增一個帳號為「test」的使用者，使用者名稱為「test」、郵件帳號為「test@test.com.tw」、密碼為「a123456!」。

輸入「http://127.0.0.1:8000/adduser/」，會新增一個帳號為「test」的使用者。



第二次執行因為該使用者已經建立，會出現使用者已經建立訊息。



2

登入和登出

7-2 登入和登出

7.2.1 登入驗證

利用auth 內建的authenticate 方法可以完成使用者登入驗證。語法：

```
user = authenticate(username= 帳號 , password= 密碼 )
```

7.2.2 登入

```
auth.login(request,user)
```

```
if user.is_active:  
    auth.login(request,user)
```

7.2.3 登出

```
auth.logout(request)
```

範例：使用者登入和登出

這個範例我們在<index.html> 首頁進行登入和登出，帳號名稱是使用User 資料表建立的資料。

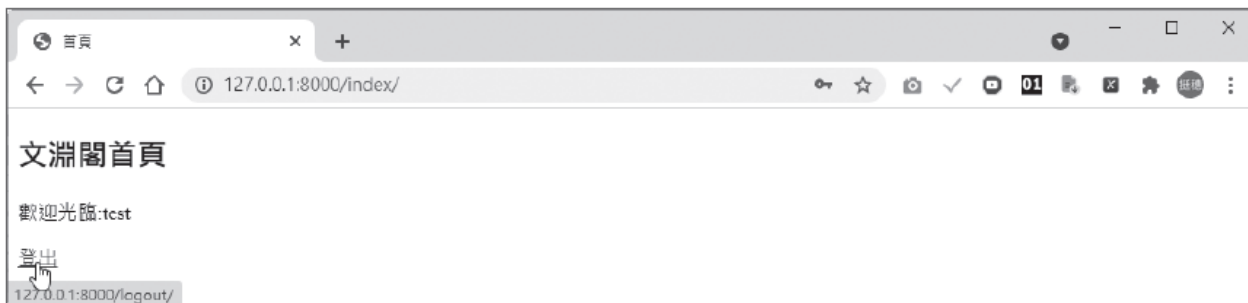
執行「127.0.0.1:8000/index/」開啟<index.html> 首頁，如果該使用者尚未通過認證，顯示「您尚未登入喔~」訊息，按 **登入** 鈕進行登入動作。



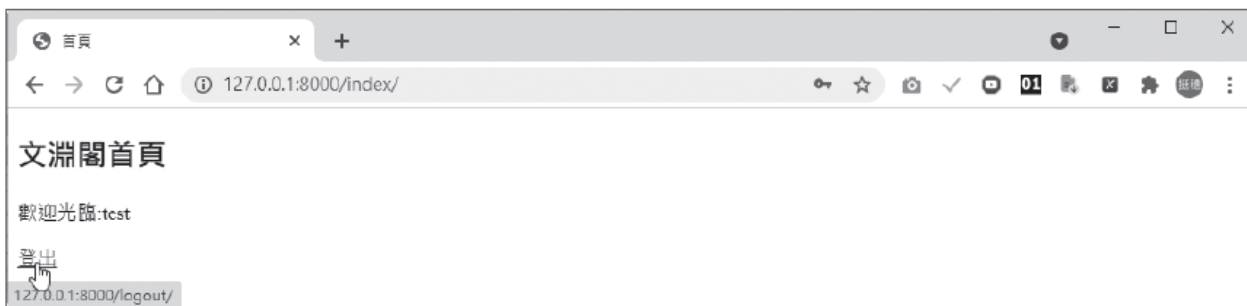
進入login 登入畫面，帳號、密碼分別輸入「test」、「a123456!」，按 **登入** 鈕。



登入成功後顯示「歡迎光臨」訊息，按 **登出** 鈕可以進行登出。



因為使用者認證通過後，會在Server 端建立Session 記錄使用者狀態，現在我們故意將瀏覽器關閉後再重新開啟，執行「127.0.0.1:8000/index/」開啟 <index.html> 首頁，首頁會顯示使用者已經認證通過，按 **登出** 鈕即可進行登出動作。





Exercise 7-1:adduser


```
urlpatterns = [  
    path('admin/', admin.site.urls),  
  
    path('', index),  
    path('index/', index),  
  
    path('login/', login),  
    path('logout/', logout),  
  
    path('adduser/', adduser),  
]
```

views.adduser

```
36 def adduser(request):|
37     try:
38         user = User.objects.get(username="test")
39     except:
40         user = None
41     if user != None:
42         message = user.username + " 帳號已建立!"
43         return HttpResponse(message)
44     else: # 建立 test 帳號
45         user = User.objects.create_user("test", "test@test.com.tw", "a123456!")
46         user.first_name = "wen" # 姓名
47         user.last_name = "lin" # 姓氏
48         user.is_staff = True # 工作人員狀態
49         user.save()
50         return redirect('/admin/')

```



Exercise 7-2: Authenticate user

```
urlpatterns = [  
    path('admin/', admin.site.urls),  
  
    path('', index),  
    path('index/', index),  
  
    path('login/', login),  
    path('logout/', logout),  
  
    path('adduser/', adduser),  
]
```

```
8  def index(request):  
9      if request.user.is_authenticated:  
10         name = request.user.username  
11         return render(request, "index.html", locals())
```

index.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
5     <title>首頁</title>
6 </head>
7 <body>
8     <h2>文淵閣首頁</h2>
9     {% if request.user.is_authenticated %}
10         歡迎光臨:{{name}}
11         <p><a href="/logout/">登出</a></p>
12     {% else %}
13         <p>您尚未登入喔~</p>
14         <a href="/login/">登入</a>
15     {% endif %}
16     <span style="color:red">{{message}}</span>
17 </body>
18 </html>
```

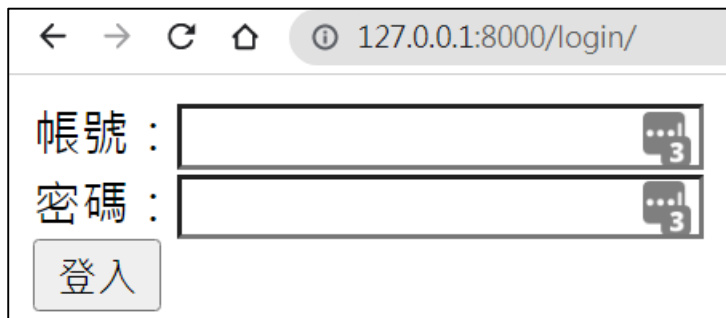
views.login

```
14 def login(request):
15     if request.method == 'POST':
16         name = request.POST['username']
17         password = request.POST['password']
18         user = authenticate(username=name, password=password)
19         if user is not None:
20             if user.is_active:
21                 auth.login(request, user)
22                 return redirect('/index/')
23                 message = '登入成功！'
24             else:
25                 message = '帳號尚未啟用！'
26         else:
27             message = '登入失敗！'
28     return render(request, "login.html", locals())
```

views.logout

```
31  def logout(request):  
32      auth.logout(request)  
33      return redirect('/index/')
```


Results

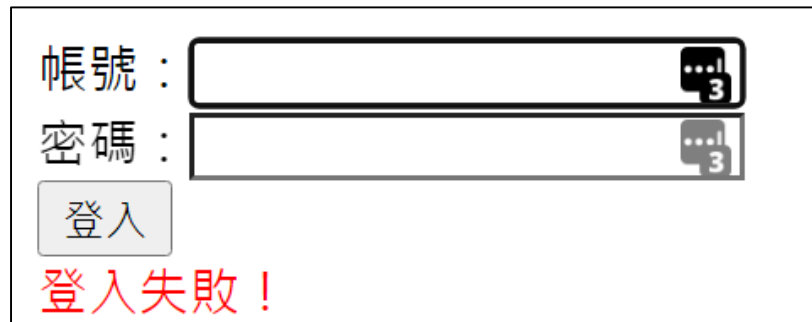


← → ↻ 🏠 ⓘ 127.0.0.1:8000/login/

帳號:

密碼:

登入



帳號:

密碼:

登入

登入失敗！



← → ↻ 🏠 ⓘ 127.0.0.1:8000/index/

文淵閣首頁

歡迎光臨:test01

[登出](#)