**使用者互動表單**

<https://ithelp.ithome.com.tw/articles/10212441>

增加{{path}}變量到html模板：

[C:\Users\小青\myrest\myrestapp\templates\menu1.html](file:///C:\Users\小青\myrest\myrestapp\templates\menu1.html)

模板上多出了一個{{path}}變量，所以要在veiws.py裡準備好要被對應的path:

[C:\Users\小青\myrest\myrestapp\views.py](file:///C:\Users\小青\myrest\myrestapp\views.py)

**重要的request屬性和方法**

|  |  |
| --- | --- |
| **屬性方法** | 說明 |
| **request.path** | /menu/ |
| **request.path\_host()** | 主機名稱(網域名稱) |
| **request.get\_full\_path()** | 請求路徑+查詢字符 |
| **request.is\_secure()** | 是否通過https協定來訪問 |

**Header訊息—meta**

1. request物件一個紀錄header訊息的字典—meta
2. meta可以把使用者的訊息或是客戶端的ip位置都查的到

增加meta到urls.py裡：[C:\Users\小青\myrest\myrest\urls.py](file:///C:\Users\小青\myrest\myrest\urls.py)

增加def meta(request)到views.py裡：[C:\Users\小青\myrest\myrestapp\views.py](file:///C:\Users\小青\myrest\myrestapp\views.py)

**數據提交訊息—GET 和 POST**

1. GET和POST是Http裡面最重要的兩個方法，這兩個方法負責提交使用者表單的數據
2. 兩者在表現上的差別在:GET方法會透過在請求路徑後面添增查詢字符來提交數據，而POST方法是隱性地傳送數據的鍵值對

|  |  |
| --- | --- |
| 方法 | 範例url |
| GET | www.restaurants.com/restaurant/?id=1 |
| POST | www.restaurants.com/restaurant |

**存在request中的提交數據**

1. POST跟GET，這兩個方法所提交的數據都會以鍵值對的方式儲存於request物件中，分別存於request.GET和request.POST

**寫一個歡迎的頁面**

**Urls.py:**[**C:\Users\小青\myrest\myrest\urls.py**](file:///C:\Users\小青\myrest\myrest\urls.py)

**Welcome.html:**[**C:\Users\小青\myrest\templates\welcome.html**](file:///C:\Users\小青\myrest\templates\welcome.html)

**\*\*(veiw會自己去找模板，所以取名很重要—盡量不要同樣名字)**

**Views.py:**[**C:\Users\小青\myrest\myrest\views.py**](file:///C:\Users\小青\myrest\myrest\views.py)

**\*\*( views裡的def 取名很重要，盡量不要跟資料夾或app同名 )**

**餐廳列表的頁面**

**Urls.py:**[**C:\Users\小青\myrest\myrest\urls.py**](file:///C:\Users\小青\myrest\myrest\urls.py)

**Restaurant\_list.html:**[**C:\Users\小青\myrest\myrestapp\templates\restaurants\_list.html**](file:///C:\Users\小青\myrest\myrestapp\templates\restaurants_list.html)

**Views.py:**[**C:\Users\小青\myrest\myrestapp\views.py**](file:///C:\Users\小青\myrest\myrestapp\views.py)

**目標：點取餐廳可以，顯示一個餐廳的菜單**

**Step1：菜單的頁面html**

Urls.py:[**C:\Users\小青\myrest\myrest\urls.py**](file:///C:\Users\小青\myrest\myrest\urls.py)

Menu3.html:[**C:\Users\小青\myrest\myrestapp\templates\menu3.html**](file:///C:\Users\小青\myrest\myrestapp\templates\menu3.html)

Views.py:[**C:\Users\小青\myrest\myrestapp\views.py**](file:///C:\Users\小青\myrest\myrestapp\views.py)

**Step2:選擇餐廳並觀看**

**方法(一)：使用表單**

**為餐廳列表加入表單:**

Html:[C:\Users\小青\myrest\myrestapp\templates\restaurants\_list.html](file:///C:\Users\小青\myrest\myrestapp\templates\restaurants_list.html)

**<td>< input type="radio" name="id" value="{{r.id}}"></td>**

Views.py:[C:\Users\小青\myrest\myrestapp\views.py](file:///C:\Users\小青\myrest\myrestapp\views.py)

**myrestapp=Myrestapp1.objects.get(id=request.GET['id'])**

**方法(二)：使用連結與get查詢字符**

Html:

餐廳的:[C:\Users\小青\myrest\myrestapp\templates\restaurants\_list1.html](file:///C:\Users\小青\myrest\myrestapp\templates\restaurants_list1.html)

**<td><a href="/menu3/?id={{r.id}}"> <3 </a></td>**

Views.py:

[C:\Users\小青\myrest\myrestapp\views.py](file:///C:\Users\小青\myrest\myrestapp\views.py)

Urls.py:

[C:\Users\小青\myrest\myrest\urls.py](file:///C:\Users\小青\myrest\myrest\urls.py)

**方法(三)：使用連結與url參數**

Html:[C:\Users\小青\myrest\myrestapp\templates\restaurants\_list2.html](file:///C:\Users\小青\myrest\myrestapp\templates\restaurants_list2.html)

**<td><a href="/menu3/{{r.id}}"> <3 </a></td>**

url.py:[C:\Users\小青\myrest\myrest\urls.py](file:///C:\Users\小青\myrest\myrest\urls.py)

**re\_path(r'menu3/(\d{1,5})',menu5),**

views.py:[C:\Users\小青\myrest\myrestapp\views.py](file:///C:\Users\小青\myrest\myrestapp\views.py)

**def menu5(request,id):**

**###有時間可以試試以class的方式去寫很像的def**

**製作餐廳的評價**

**(以restaurant\_list1去製作)**

Html:

(restaurant\_list1):[C:\Users\小青\myrest\myrestapp\templates\restaurants\_list1.html](file:///C:\Users\小青\myrest\myrestapp\templates\restaurants_list1.html)

(comment):[C:\Users\小青\myrest\myrestapp\templates\comment.html](file:///C:\Users\小青\myrest\myrestapp\templates\comment.html)

Urls.py: [C:\Users\小青\myrest\myrest\urls.py](file:///C:\Users\小青\myrest\myrest\urls.py)

Views.py: [C:\Users\小青\myrest\myrestapp\views.py](file:///C:\Users\小青\myrest\myrestapp\views.py)

**\*\*在views.py裡增加的Comment(原本在myrestapp.models裡的)要記得import**

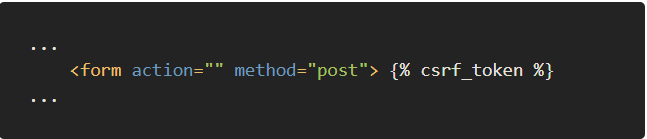
Admin.py:[C:\Users\小青\myrest\myrestapp\admin.py](file:///C:\Users\小青\myrest\myrestapp\admin.py)

Models.py:[C:\Users\小青\myrest\myrestapp\models.py](file:///C:\Users\小青\myrest\myrestapp\models.py)

**\*\*admin裡的時間(現在時間不太對，之後再研究)**

**CSRF(cross-site request forgery)**

**Django幫我們啟動了CSRF攻擊的防護  
CSRF(cross-site request forgery)是惡意的跨站請求或偽裝使用者的攻擊  
攻擊者會欺騙用戶的瀏覽器去訪問一個認證過的網站並且執行一些惡意的操作  
由於用戶的瀏覽器已經被該網站認證過了，所以該網站會放心的讓這些操作被執行  
即便這些操作並非該網站要求的或是不是用戶自願的  
所以伺服器需要有一些保護措施  
常見的防護手段就是使用一個伺服器產生的亂數token，夾帶在送給用戶端的表單中  
當用戶端送回表單時，伺服器檢查這個token是不是自己發出，便可以防止攻擊  
因為在settings.py中的MIDDLEWARE有預設的'django.middleware.csrf.CsrfViewMiddleware'  
所以Django會要求CSRF token認證  
在comments.html中的表單加入{% csrf\_token %}**

****