**權限與註冊(c14)**

<https://ithelp.ithome.com.tw/articles/10212466>

1. 讓HttpRequest物件的user屬性進行is\_authenticated判斷後🡪匿名用戶
2. 有些頁面不給造訪🡪

|  |  |
| --- | --- |
| 方法(一)  在html裡引導使用者的去向 | 無法防止使用者直接輸入網址去到不讓去的網頁 |
| 方法(二)  在views.py裡面將匿名用戶引導到login頁面  他login成功後再讓他進該網頁 | 需要對views.py裡每個需要判定登入的def()寫**檢查碼**和**重導**  [C:\Users\小青\myrest\myrestapp\views.py](file:///C:\Users\小青\myrest\myrestapp\views.py)  **\*\*\*無法讓後來才登入的直接進到restaurant\_list1 裡\*\*\***  **不理解為何會跑到/index/去** |
| 方法(三)  使用**login\_required**修飾符 | **from django.contrib.auth.decorators import login\_required**  在要寫檢查和重導的def()上加上裝飾器：**@login\_required**  [C:\Users\小青\myrest\myrestapp\views.py](file:///C:\Users\小青\myrest\myrestapp\views.py)  (或者加在urls.py裡也是一樣的效果)  **\*\*\*無法讓後來才登入的使用者直接進到restaurant\_list 裡\*\*\***  **不理解為何會跑到/index/去** |
| 結合方法(一)和方法(三) | 讓使用者方便也讓寫程式的人輕鬆  (再想想) |

**註冊**

**建立新用戶：**

先進入django shell : **python manage.py shell**

**from django.contrib.auth.models import User**

**new\_user=User.objects.create(username=”peter”,password=”test”)**

這樣就建立好了

**更改用戶密碼：**

**new\_user.set\_password(“hello123”)**

**new\_user.is\_staff=True** 🡪設置為True可讓該使用者登入admin後台

**new\_user.save()**

**\*\*密碼的加密django使用的是加入隨機值的hash演算法\*\***

**(可再看看相關資料)**

**註冊用戶：**

1. 處理註冊的視圖函式：

* [C:\Users\小青\myrest\myrest\views.py](file:///C:\Users\小青\myrest\myrest\views.py)
* 使用django 中auth應用內建的註冊表單模型：UserCreationForm
* UserCreationForm是繼承自forms.ModelForm的表單模型
* **ModelForm**:

是一種特殊形式的表單，通常在表單的欄位和某資料庫模型的欄位相同時，用ModelForm可以減少一些不必要的手續。

從ModelForm產生一個資料庫模型的物件並且將之存入資料庫的步驟：

1. 填入欄位資料並創造表單物件
2. 使用save方法以表單中的內容生成資料並存入資料庫

* **資料庫模型要產生實例**時，參數使用的是以各欄位名稱為名的關鍵字引數

**表單模型產生實例**時，使用的是字典型態的引數

1. 一個註冊用的頁面：

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

1. 用來註冊的表單：

* 使用form的as\_p方法來生成表單
* Html:[C:\Users\小青\myrest\templates\registor.html](file:///C:\Users\小青\myrest\templates\registor.html)

**權限(Permission)**

**建立權限**

1. 在具名用戶裡區分不同等級的用戶，並給予不同的權限。
2. **Permission** : django內建的模型

|  |  |
| --- | --- |
| Django主要包含的欄位名稱 | 說明 |
| name | 權限名稱 |
| codename | 實際運用在代碼中的權限名稱 |
| Content\_type | * 來自一個模型的content\_type * 每一個權限都會跟一個資料庫模型綁定，都會屬於一種資料庫模型，這也是需要content\_type的原因 |

1. 新增權限的方法(有兩種)：

* (第一種)透過資料庫模型中的Meta Class來設定

Example: 為餐廳評論新增一個權限

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

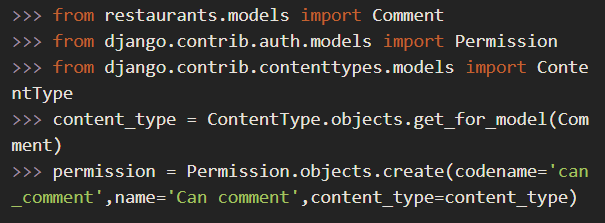
這個方法的context\_type由django自動取得不須定義

1. 由於權限也是模型，也有資料庫，所以必須同步模型：



* (第二種)透過操作permission模型來建立新物件

Python manage.py shell



為了取得content\_type，使用get\_for\_model方法，以一個要綁定的模型作為參數

* 新增了權限後，可以到admin頁面選擇想新增的權限種類

(之後再試試看新增、移除權限功能)

**用shell新增、移除權限的方法與判定(未實作)**



用**User.user\_permissions.add**和**User.user\_permisssions.remove**來為某個使用者新增或刪除一個權限，也可以用User.has\_perm來查看使用者是否具備某種權限

只是如果沒有向管理器重新取得User物件的話，has\_perm無法反應出即刻性的結果

這並不奇怪，因為Django對於增加或刪除權限的動作都會直接作用在資料庫上  
被取出來的實例反而不受影響，所以我們有必要重新取得user物件

用戶權限的幾個常用方法:

|  |  |  |
| --- | --- | --- |
| 方法 | 說明 | 範例 |
| add | 讓用戶新增權限  參數，是一個或多個Permission物件 | **User.user\_permissions.add(perm)** |
| remove | 讓用戶刪除權限  參數，是一個或多個Permission物件 | **User.user\_permission.remove(perm)** |
| has\_perm | 確認某用戶是否具備某種權限  參數，是一個字串，形式為<應用名稱>.<權限的codename> | **User.has\_perm(‘myrest.can\_comment’)** |
| clear | 清除用戶的所有權限 | **User.user\_permissions.clear()** |

**Django自帶權限**

如果使用了Django的auth應用，則Django會自動對專案中，已安裝應用底下所有的模型創建以下三種權限:

1.add權限

2.change權限

3.delete權限

**使用權限(之後實測後補齊)**

對於不同權限的用戶之間要如何做出差異化有以下手法(有些頁面不給訪問):

|  |  |
| --- | --- |
| 方法(一)  在html裡引導使用者的去向 | 無法防止使用者直接輸入網址去到不讓去的網頁  要搭配的views.py: |
| 方法(二)  在views.py裡面將匿名用戶引導到login頁面  他login成功後再讓他進該網頁 | 需要對views.py裡每個需要判定登入的def()寫**檢查碼**和**重導**  [C:\Users\小青\myrest\myrestapp\views.py](file:///C:\Users\小青\myrest\myrestapp\views.py)  **\*\*\*無法讓後來才登入的直接進到restaurant\_list1 裡\*\*\***  **不理解為何會跑到/index/去** |
| 方法(三)  使用**login\_required**修飾符 | View.py: [C:\Users\小青\myrest\myrestapp\views.py](file:///C:\Users\小青\myrest\myrestapp\views.py)  **from django.contrib.auth.decorators import permission\_required**  在要寫檢查和重導的def()上加上裝飾器：**@permission\_required**  (本來是要自己寫  **from django.contrib.auth.decorators import user\_passes\_test**  **def user\_can\_comment(user): …**  因為是一種常態所以才發展成@permission\_required)  [C:\Users\小青\myrest\myrestapp\views.py](file:///C:\Users\小青\myrest\myrestapp\views.py)  (或者加在urls.py裡也是一樣的效果)  **\*\*\*無法讓後來才登入的使用者直接進到restaurant\_list 裡\*\*\***  **不理解為何會跑到/index/去** |
| 結合方法(一)和方法(三) | 讓使用者方便也讓寫程式的人輕鬆  (再想想) |

**群組與群組權限**

群組的權限屬性 ： permissions

常用的群組方法：

|  |  |  |
| --- | --- | --- |
| 方法 | 說明 | 範例 |
| add | 讓用戶加入群組  參數是一或多個群組 | User.groups.add(group) |
| remove | 讓用戶離開群組  參數是一或多個群組 | User.groups.remove(group) |
| clear | 讓用戶離開所有群組 | User.groups.clear() |

****