

網站表單的應用

1

學習目標

- ▶ 網站與表單接收表單資料儲存於資料庫中
- ▶ 基礎表單類別的應用
- ▶ 模型表單類別的應用

HTML <form>表單簡介

► 以下是一個非常典型的表單HTML網頁設計例：

- <form name='my form' action='/' method='GET'>
- <label for='user_id'>Your ID:</label>
- <input id='user_id' type='text' name='user_id'>
- <label for='user_pass'>Your Password:</label>
- <input id='user_pass' type='password' name='user_pass'>
- <input type='submit' value='登入'>
- <input type='reset' value='清除重填'>
- </form>

The screenshot shows a web browser window with the following details:

- Address bar: 檔案 | F:/NTU%20CSIE%20Train/錄影/ex/WDP312/P07/form01.html
- Form fields:
 - Your ID: (empty input field)
 - Your Password: (empty input field)
- Buttons:
 - 登入 (Login) - blue button
 - 清除重填 (Clear/Reset) - grey button

HTML <form>表單簡介

▶ 可以看出以下幾個重點：

- ▶ 以<form></form>作為開始和結束的標記
- ▶ 在<form>設定相關屬性
 - ▶ name是通用屬性，幾乎在每一個標記中都可以使用，代表本標記的名稱
 - ▶ id也是通用屬性，每一個屬性都可以使用，代表本標記的識別字，通常必須要是唯一的
 - ▶ method適用於<form>標記，用來表示傳遞的參數要使用POST還是GET，這是表單傳遞的兩種不同方式。
 - ▶ action屬性很重要，後面指定的內容是用來設定當使用者按下「submit」按鈕之後，所有的資料要被送到哪裡。一般可以指定一個PHP程式或是javascript的一個函數，在Django只要指定給一個要處理的網址就可以了。如果沒有任何設定，則從哪裡來就回哪裡去。
 - ▶ <label>主要是用來設定表單元素前的說明文字，使用for屬性來設定此標籤是屬於哪一個輸入元素之用。

HTML <form>表單簡介

- ▶ <input>為主要輸入元素，其屬性主要包括：
 - ▶ id是通用屬性，識別字
 - ▶ name是通用屬性，是本<input>的名稱，在Django程式中是以此名稱來取得資料
 - ▶ type是格式，有許多種型態，其中text為文字型態，而password 本上也是文字型態，但是此欄位在輸入時，字元會以密碼符號代替。
 - ▶ 如欲設定預設值，可使用value屬性
- ▶ <input type='submit'>為特殊元素，就是「送出資料」的按鈕，但是按鈕上的文字是由value屬性來指定。
- ▶ <input type='reset'>為特殊元素，就是「清除本表單中目前所有輸入的值」，一般都被當做是清除重來按鈕。

HTML <form>表單簡介

- ▶ 在Django的網站中，這些內容要放在Template檔案中
- ▶ 開始建立網站
 - ▶ 使用`django-admin startproject HTML_Form01` 建立一個 `HTML_Form` 網站專案
 - ▶ 使用`python manage.py startapp mysite`，建立一個名稱 `mysite` 的App
 - ▶ 將 `mysite` 加入至到 `settings.py` 中 `INSTALLED_APP`
 - ▶ 參考前面章節內容，設定 `templates` 和 `STATICFILES_DIRS` 的目錄
 - ▶ 借用上一堂課的 `templates` 目錄下的 `base.html`, `header.html`, `footer.html` 以及 `index.html`，還有 `static` 下的 `images/logo.png`

HTML <form>表單簡介

- ▶ 修改index.html如下：

```
▶ <!-- index.html (HTML_Form01) -->
▶ {% extends "base.html" %}
▶ {% block title %}我有話要說{% endblock %}
▶ {% block content %}
▶   <div class='container'>
▶     <form name='my form' action='/' method='GET'>
▶       <label for='user_id'>Your ID:</label>
▶       <input id='user_id' type='text' name='user_id'>
▶       <label for='user_pass'>Your Password:</label>
▶       <input id='user_pass' type='password' name='user_pass'>
▶       <input type='submit' value='登入'>
▶       <input type='reset' value='清除重填'>
▶     </form>
▶   </div>
▶ {% endblock %}
```

Your ID:

Your Password:

► `settings.py`

► `STATIC_URL = '/static/'`

► `STATICFILES_DIRS = [`

► `os.path.join(BASE_DIR, 'static'),`

► `]`

► `urls.py`

► `path("", views.index),`

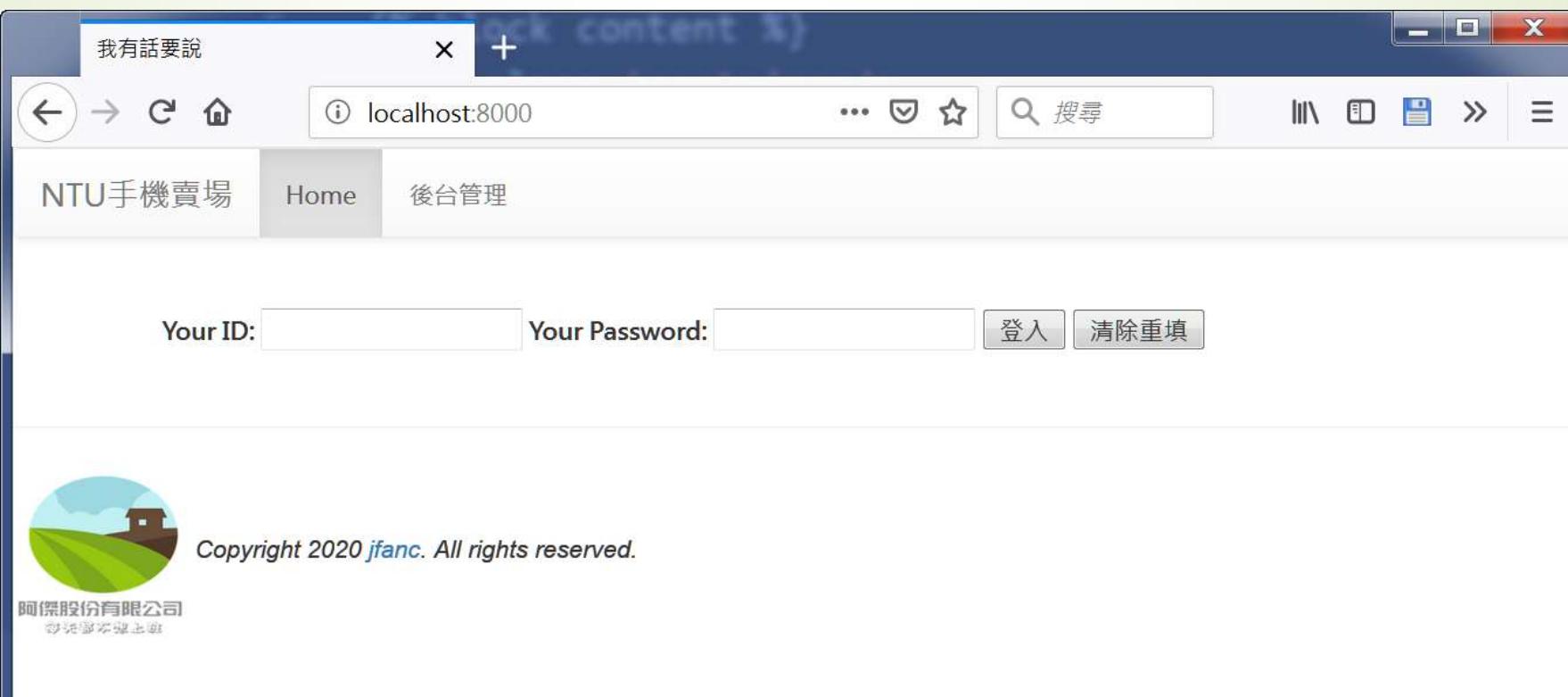
► `views.py`

► `def index(request):`

► `return render(request, 'index.html', locals())`

HTML <form>表單簡介

- ▶ 別忘了要在 `urls.py` 設定 `index` 的首頁對應以及使用 `python manage.py runserver` 執行網站
- ▶ 典型登入用表單的呈現畫面



HTML <form>表單簡介

- ▶ 如何接收輸入的值呢？
 - ▶ 是透過request物件
 - ▶ 在處理的views.index函數中會接收一個request物件，使用request.GET['input_name']取得表單資料
- ▶ views.py程式如下：
 - ▶ from django.shortcuts import render
 - ▶ from mysite import models
 - ▶ from django.template.loader import get_template
 - ▶ def index(request):
 - ▶ template = get_template('index.html')
 - ▶ try:
 - ▶ urid = request.GET['user_id']
 - ▶ urpass = request.GET['user_pass']
 - ▶ except:
 - ▶ urid = None
 - ▶ return render(request, 'index.html', locals())

HTML <form>表單簡介

- 在index.html中取出verified這個變數來檢查，依據內容是True還是False來顯示相對應的字串。修改後的index.html如下所示：

```
► <!-- index.html (HTML_Form01) -->
► {% extends "base.html" %}
► {% block title %}我有話要說{% endblock %}
► {% block content %}
► <div class='container'>
►
► <form name='my form' action='/' method='GET'>
►   <label for='user_id'>Your ID:</label>
►   <input id='user_id' type='text' name='user_id'>
►   <label for='user_pass'>Your Password:</label>
►   <input id='user_pass' type='password' name='user_pass'>
►   <input type='submit' value='登入'>
►   <input type='reset' value='清除重填'>
► </form>
► Your ID:{{ urid | default:"未輸入ID" }}<br/>
► {% if verified %}
►   <em>你通過了驗證</em>
► {% else %}
►   <em>密碼或帳號打錯了喔</em>
► {% endif %}
► </div>
► {% endblock %}
```

我有話要說

localhost:8000

NTU手機賣場 Home 後台管理

Your ID: Your Password: 登入 清除重填

Your ID:未輸入ID
密碼或帳號打錯了喔


Copyright 2020 *jfanc*. All rights reserved.

阿傑股份有限公司
你我都希望上班

13

我有話要說

localhost:8000/?user_id=123&user_pass=456

NTU手機賣場 Home 後台管理

Your ID: Your Password: 登入 清除重填

Your ID:123
密碼或帳號打錯了喔


Copyright 2020 *jfanc*. All rights reserved.

阿傑股份有限公司
臺灣第一大上架

HTML <form>表單簡介

- ▶ 表單的內容有可能會是空值 (None)
- ▶ 使用try/except例外處理機制預防造成網站程式異常中斷執行

```
▶ def index(request):  
    ▶     try:  
    ▶         urid = request.GET['user_id']  
    ▶         urpass = request.GET['user_pass']  
    ▶     except:  
    ▶         urid = None  
    ▶     if urid != None and urpass == '12345':  
    ▶         verified = True  
    ▶     else:  
    ▶         verified = False  
    ▶ return render(request, 'index.html', locals())
```

我有話要說

localhost:8000/?user_id=abc&user_pass=12345

NTU手機賣場 Home 後台管理

Your ID: abc Your Password: ●●●●● 登入 清除重填

Your ID:abc
你通過了驗證


Copyright 2020 [jfanc](#). All rights reserved.

阿傑股份有限公司
每天都希望上班

活用表單的標籤

► HTML表單中常用的元件 <https://jsbin.com/liyodimeka/edit?html,output>

標籤名稱	說明	使用範例
<select></select>	下拉式選單	<pre><label for='flist'>最喜歡的水果</label> <select name='flist'> <option value='0'>Apple</option> <option value='1'>Banana</option> <option value='2'>Cherry</option> </select>
</pre>

您使用過哪家業者的門號？(可複選)

中華電信
台灣大哥大
遠傳
威寶

提交 重新輸入



活用表單的標籤

```
<input type='radio'>
```

選項按鈕(單選)

最喜歡的顏色(單選)


```
<input type='radio' name='fcolor' value='Green'  
checked>Green<br>
```

```
<input type='radio' name='fcolor'  
value='Blue'>Blue<br>
```

```
<input type='radio' name='fcolor'  
value='Red'>Red<br>
```

```
<input type='radio' name='fcolor'  
value='Black'>Black<br>
```

最喜歡的顏色(單選)

Green

Blue

Red

Black

姓 名 :

E-Mail :

年 齡 : 未滿20歲 20~29 30~39 40~49 50歲以上

提交

重新輸入

活用表單的標籤

<https://jsbin.com/vasavifaha/edit?html,output>

<input type='checkbox'>	核取方塊（多選）	最喜歡的顏色(複選) <input type='checkbox' name='cfcolor' value='Green'>Green <input type='checkbox' name='cfcolor' value='Blue'>Blue <input type='checkbox' name='cfcolor' value='Red'>Red <input type='checkbox' name='cfcolor' value='Black'>Black
--------------------------------------	----------	--

您最喜歡的運動是:

- 游泳
- 跑步
- 打籃球
- 高爾夫
- 有氧舞蹈

checkbox測試



最喜歡的顏色(複選)

- Green
- Blue
- Red
- Black

活用表單的標籤

<code><input type='hidden'></code>	隱藏欄位	<code><input type='hidden' name='hidevalue' value='hidevalue'></code>
<code><input type='button'></code>	自訂按鈕	<code><input type='button' value='Google' onclick='location.href="http://google.com"'></code>
<code><textarea></textarea></code>	多列文字內容	<code><textarea name='message' rows=5 cols=40></textarea></code>

大哥大使用意見調查

姓 名 :

E-Mail :

活用表單的標籤

- ▶ 提供西元年的下拉式選單功能，希望能從西元1960年到2020年
- ▶ `views.index`中加入如下程式碼：
 - ▶ `years = range(1960,2021)`
 - ▶ 在`index.html`中使用以下的內容把`years`串列轉換成下拉式選單：
 - ▶ `<label for='byear'>出生年:</label>`
 - ▶ `<select name='byear'>`
 - ▶ `{% for year in years %}`
 - ▶ `<option value='{{ year }}'>{{ year }} </option>`
 - ▶ `{% endfor %}`
 - ▶ `</select>
`

我有話要說

localhost:8000/?user_id=abc&user_pass=12345

NTU手機賣場 Home 後台管理

出生年: 1960 ▾

Your ID: Your Password: 登入 清除重填

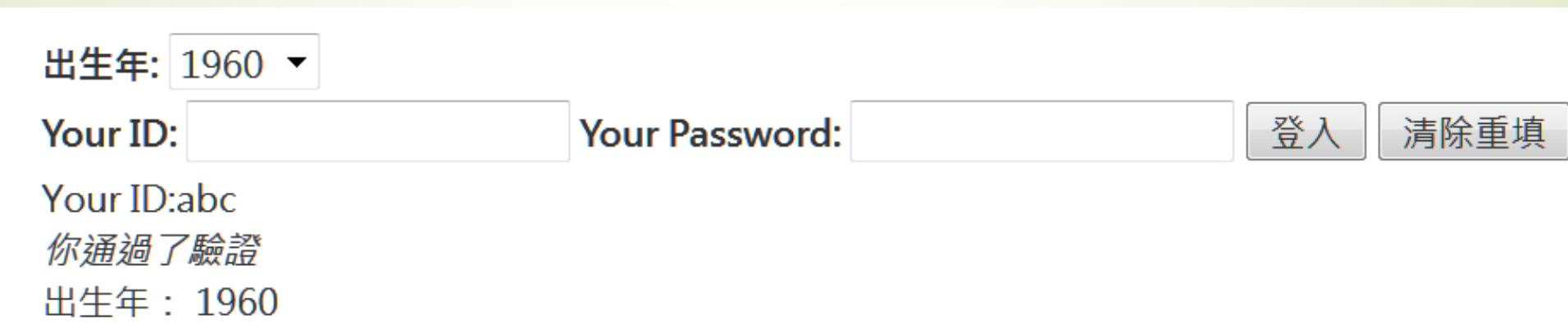
Your ID:abc
你通過了驗證


Copyright 2020 [jfanc](#). All rights reserved.

阿傑股份有限公司
全球服務上線

活用表單的標籤

- ▶ 下拉式選單只有**單選**，所以在views.index中使用
`se_byear = request.GET['byear']`
可以取得上述下拉式選單的結果
- ▶ 在index.html中新增
- ▶ 出生年：`{{se_byear}}`



The screenshot shows a user interface for a login page. At the top left is a dropdown menu labeled "出生年" with the value "1960". Below it are two input fields: "Your ID:" containing "Your ID:abc" and "Your Password:". To the right of these fields are two buttons: "登入" (Login) and "清除重填" (Clear/Reset). Below the input fields, there is a message "你通過了驗證" (You have passed verification) and at the bottom, the text "出生年：1960".

► checkbox這一類多選的選單，則要以串列變數的方式來處理，
例如以下的HTML片段：

►喜歡的顏色（可複選）：

- <input type='checkbox' name='fcolor' value='Green'>Green
- <input type='checkbox' name='fcolor' value='Red'>Red
- <input type='checkbox' name='fcolor' value='Blue'>Blue
- <input type='checkbox' name='fcolor' value='Yellow'>Yellow
- <input type='checkbox' name='fcolor' value='Orange'>Orange
-

活用表單的標籤

- ▶ 顏色選項的name屬性均為相同，但是value不一樣，使用getlist方法函數可以把所有選到的項目放到串列中
- ▶ 在views.index函數中要使用下列這一行程式來取得：
 - ▶ urfcolor = request.GET.getlist('fcolor')
- ▶ {% for %}/{% endfor %}作法將urfcolor串列變數在index.html中顯示出來，如下所示：
 - ▶ 顏色喜好:
 - ▶ {% for c in urfcolor %}
 - ▶ {{ c }}
 - ▶ {% empty %}
 - ▶ 沒有選擇任何顏色
 - ▶ {% endfor %}
 - ▶

活用表單的標籤

► 加上各式HTML表單標籤的網頁執行畫面



小練習

- 依照上面的方法
- 新增一個 radio 選單
- 選擇一部喜歡的電影

出生年:

Green Red Blue Yellow Orange

喜好的電影 哥吉拉2 阿拉丁 你的名字 神隱少女 Evangelion

Your ID: Your Password:

Your ID:abc
密碼或帳號打錯了喔
出生年：1960
顏色喜好: Red Yellow
喜歡的電影：阿拉丁

建立本堂課範例網站的資料模型

- ▶ 延續上一堂課的Template基礎，設計一個可以讓網友自由發言的網站，功能如下：
 - ▶ 可以自選心情
 - ▶ 張貼心情小語
 - ▶ 設定一個日後要刪除文章用的密碼
 - ▶ 訊息者必須經過管理者從後台啟用之後才會顯示在網頁上
(文章有一個布林值欄位enabled，預設值是False，只有開啟為True之後，才允許出現在網頁中)

建立本堂課範例網站的資料模型

► 我有話要說網站的主畫面設計



建立本堂課範例網站的資料模型

► Model.py類別設計：

► class Mood

► status - 一個文字欄位，用來記錄心情的狀態

► class Post

► mood - 連結到Mood使用外鍵連結過去

► nickname - 記錄張貼者暱稱

► message - 儲存實際的內容

► del_pass - 記錄可以刪除此篇訊息的密碼

► pub_time - 設定為自動填入的修改時間

► enabled - 預設是False，決定是否要把這筆資料顯示在網頁上

建立本堂課範例網站的資料模型

- ▶ models.py的內容如下所示：

```
▶ from django.db import models  
  
▶ class Mood(models.Model):  
    status = models.CharField(max_length=10, null=False)  
  
    def __str__(self):  
        return self.status  
  
▶ class Post(models.Model):  
    mood = models.ForeignKey('Mood', on_delete=models.CASCADE)  
    nickname = models.CharField(max_length=10, default='不願意透漏身份的人')  
    message = models.TextField(null=False)  
    del_pass = models.CharField(max_length=10)  
    pub_time = models.DateTimeField(auto_now=True)  
    enabled = models.BooleanField(default=False) #改True就可以預設為開啟  
  
    def __str__(self):  
        return self.message
```

建立本堂課範例網站的資料模型

- ▶ 上述Model經由makemigrations和migrate同步到網站資料庫之後，可以到admin.py中再加入下列程式碼：
 - ▶ from django.contrib import admin
 - ▶ from mysite import models

- ▶ class PostAdmin(admin.ModelAdmin):
 - ▶ list_display=('nickname', 'message', 'enabled', 'pub_time')
 - ▶ ordering=(-'pub_time',)
- ▶ admin.site.register(models.Mood)
- ▶ admin.site.register(models.Post, PostAdmin)

建立本堂課範例網站的資料模型

► 在views.index處理函數中把這些資料都先讀取出來，放在各個變數中備用，如下所示：

```
► # -*- encoding:utf-8 -*-
► from django.template.loader import get_template
► from django.http import HttpResponse
► from mysite import models

► def index(request):
►     template = get_template('index.html')
►     posts = models.Post.objects.filter(enabled=True).order_by(
►         '-pub_time')[:30]
►     moods = models.Mood.objects.all()
►     return render(request, 'index.html', locals())
```

網站表單的建立與資料顯示

- ▶ `index.html`的內容，先設計表單的部份：

- ▶ `<form name='my form' action='/' method='GET'>`
- ▶ 現在的心情：`
`
- ▶ `{% for m in moods %}`
- ▶ `<input type='radio' name='mood' value='{{ m.status }}'>{{ m.status }}`
- ▶ `{% endfor %}`
- ▶ `
`
- ▶ 心情留言板：`
`
- ▶ `<textarea name='user_post' rows=3 cols=70></textarea>
`
- ▶ `<label for='user_id'>你的暱稱：</label>`
- ▶ `<input id='user_id' type='text' name='user_id'>`
- ▶ `<label for='user_pass'>張貼密碼：</label>`
- ▶ `<input id='user_pass' type='password' name='user_pass'>
`
- ▶ `<input type='submit' value='張貼'>`
- ▶ `<input type='reset' value='清除重填'>`
- ▶ `</form>`

我有話要說 + posts = models.Post.objects.filter(enabled=True).order

localhost:8000 110% 搜尋

NTU手機賣場 Home 後台管理

現在的心情：

心情留言板：

你的暱稱： 張貼密碼：


阿傑股份有限公司
夢想都不要上班

Copyright 2020 jfanc. All rights reserved.

► 先 createsuperuser

新增心情種類

► 登入後台後新增心情如下：

The screenshot shows the Django administration interface for the 'Moods' model. The title bar reads '我有話要說' and 'Select mood to change | Django'. The browser address bar shows 'localhost:8000/admin/mysite/mood/'. The main content area is titled 'Django administration' and 'WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT'. Below this, the breadcrumb navigation shows 'Home > Mysite > Moods'. The page title is 'Select mood to change'. On the right, there is a button labeled 'ADD MOOD +' with a plus sign. On the left, there is a 'Action:' dropdown menu and a 'Go' button. The list of moods is displayed in a table format:

<input type="checkbox"/> MOOD
<input type="checkbox"/> 覺得暗爽
<input type="checkbox"/> 覺得難過
<input type="checkbox"/> 覺得開心
<input type="checkbox"/> 很差
<input type="checkbox"/> 很棒

At the bottom of the list, it says '5 moods'.

我有話要說 x + posts = models.Post.objects.filter(enabled=True).order

localhost:8000 110% 搜尋

NTU手機賣場 Home 後台管理

現在的心情：

很棒 很差 覺得開心 覺得難過 覺得暗爽

心情留言板：

你的暱稱： 張貼密碼：


阿傑股份有限公司
夢想都不要上班

Copyright 2020 jfanc. All rights reserved.

網站表單的建立與資料顯示

► 利用Bootstrap的Grid網格系統以及Panel功能設定置中的訊息：

- <div class='row'>
- <div class='col-md-12'>
- <div class='panel panel-default'>
- <div class='panel-heading' align=center>
- <h3>~~寶寶心裡苦，寶寶只在這裡說~~</h3>
- </div>
- </div>
- </div>
- </div>

接收表單資料儲存於資料庫中

- ▶ 修改views.index處理函數如下所示：

```
▶ def index(request):  
▶     posts = models.Post.objects.filter(enabled = True).order_by('-pub_time')[:30]  
▶     moods = models.Mood.objects.all()  
▶     try:  
▶         user_id = request.GET['user_id']  
▶         user_pass = request.GET['user_pass']  
▶         user_post = request.GET['user_post']  
▶         user_mood = request.GET['mood']  
▶     except:  
▶         user_id = None  
▶         message = '如要張貼訊息，則每一個欄位都要填...'  
  
▶     if user_id != None:  
▶         mood = models.Mood.objects.get(status=user_mood)  
▶         post = models.Post.objects.create(mood=mood, nickname=user_id, del_pass=user_pass, message=user_post)  
▶         post.save()  
▶         message='成功儲存！請記得你的編輯密碼[{}]！，訊息需經審查後才會顯示。'.format(user_pass)  
  
▶     return render(request, 'index.html', locals())
```

接收表單資料儲存於資料庫中

► 在index.html中，我們只有在表單之前新增了一個顯示訊息的程式碼，如下所示：

```
► {%
    if message %}

    ► <div class='alert alert-warning'>
        ► {{ message }}
    ► </div>

► {%
    endif %}
```

我有話要說

Site administration | Django site: + request.GET['mood']

localhost:8000/?user_post=123&user=

110% ... 🔍 ⭐ 🔍 搜尋

NTU手機賣場 Home 後台管理

現在的心情：

心情留言板：

你的暱稱： 張貼密碼：

張貼 清除重填

~~寶寶心裡苦，寶寶只在這裡說~~

如要張貼訊息，則每一個欄位都要填...

 Copyright 2020 jfanc. All rights reserved.

我有話要說 × Select mood to change | Django × +

localhost:8000/?user_post=123&user_id=110% ... 搜尋

NTU手機賣場 Home 後台管理

現在的心情：

很棒 很差 覺得開心 覺得難過 覺得暗爽

心情留言板：

買到超便宜的手機

你的暱稱：富二代 張貼密碼：●●●●●

張貼 清除重填

~~寶寶心裡苦，寶寶只在這裡說~~

如要張貼訊息，則每一個欄位都要填...


Copyright 2020 jfanc. All rights reserved.

國際股份有限公司
台灣網上書店

我有話要說 x Select mood to change | Django x +

localhost:8000/?mood=覺得暗爽&user_pc=110% ... 🔍 ⚡ ⚡ 🔍 搜索

NTU手機賣場 Home 後台管理

現在的心情：

很棒 很差 覺得開心 覺得難過 覺得暗爽

心情留言板：

你的暱稱： 張貼密碼：

張貼 清除重填

~~寶寶心裡苦，寶寶只在這裡說~~

成功儲存！請記得你的編輯密碼[123456]！· 訊息需經審查後才會顯示。


Copyright 2020 jfanc. All rights reserved.

同儕股份有限公司
尊榮的本屋上網



Django administration

WELCOME, ADMIN. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home > Mysite > Posts

Select post to change

ADD POST +

Action: -----

Go

0 of 1 selected

<input type="checkbox"/>	NICKNAME	MESSAGE	ENABLED	PUB TIME
<input type="checkbox"/>	富二代	買到超便宜的手機	✖	June 21, 2019, 7:21 p.m.

1 post

網站表單的建立與資料顯示

► 顯示訊息內容的程式碼：

```
► <div class="row">  
► {%- for p in posts %}  
►   <div class="col-sm-12 col-md-4">  
►     <div class='panel panel-primary'>  
►       <div class='panel-heading'>【{{ p.nickname }}】覺得{{ p.mood }}</div>  
►       <div class='panel-body'>{{ p.message | linebreaks }}</div>  
►       <div class='panel-footer' align='right'>  
►         <i><small>{{ p.pub_time }}</small></i>  
►       </div>  
►     </div>  
►   </div>  
► {%- endfor %}  
► </div>
```

新增顯示貼文

- ▶ `views.py`
- ▶ `posts = models.Post.objects.filter(enabled=True).order_by('-pub_time')[:150]`

我有話要說

Select post to change | Django site

localhost:8000/admin/mysite/post/

WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT

Django administration

Home > Mysite > Posts

The post "我的鑽石掉了" was changed successfully.

Select post to change

Action: ----- Go 0 of 3 selected

<input type="checkbox"/>	NICKNAME	MESSAGE	ENABLED	PUB TIME
<input type="checkbox"/>	孫小美	我的鑽石掉了	✖	June 21, 2019, 7:59 p.m.
<input type="checkbox"/>	阿土伯	今天在田裡挖到鑽石	✓	June 21, 2019, 7:59 p.m.
<input type="checkbox"/>	富二代	買到超便宜的手機	✓	June 21, 2019, 7:49 p.m.

3 posts

我有話要說 Select post to change | Django select post to change

localhost:8000 110% 搜索

現在的心情：
◎很棒 ◎很差 ◎覺得開心 ◎覺得難過 ◎覺得暗爽

心情留言板：

你的暱稱： 張貼密碼：

張貼 清除重填

~~寶寶心裡苦，寶寶只在這裡說~~

【富二代】覺得覺得暗爽
買到超便宜的手機
June 21, 2019, 7:49 p.m.

【sdf】覺得覺得難過
sdf
June 21, 2019, 7:48 p.m.

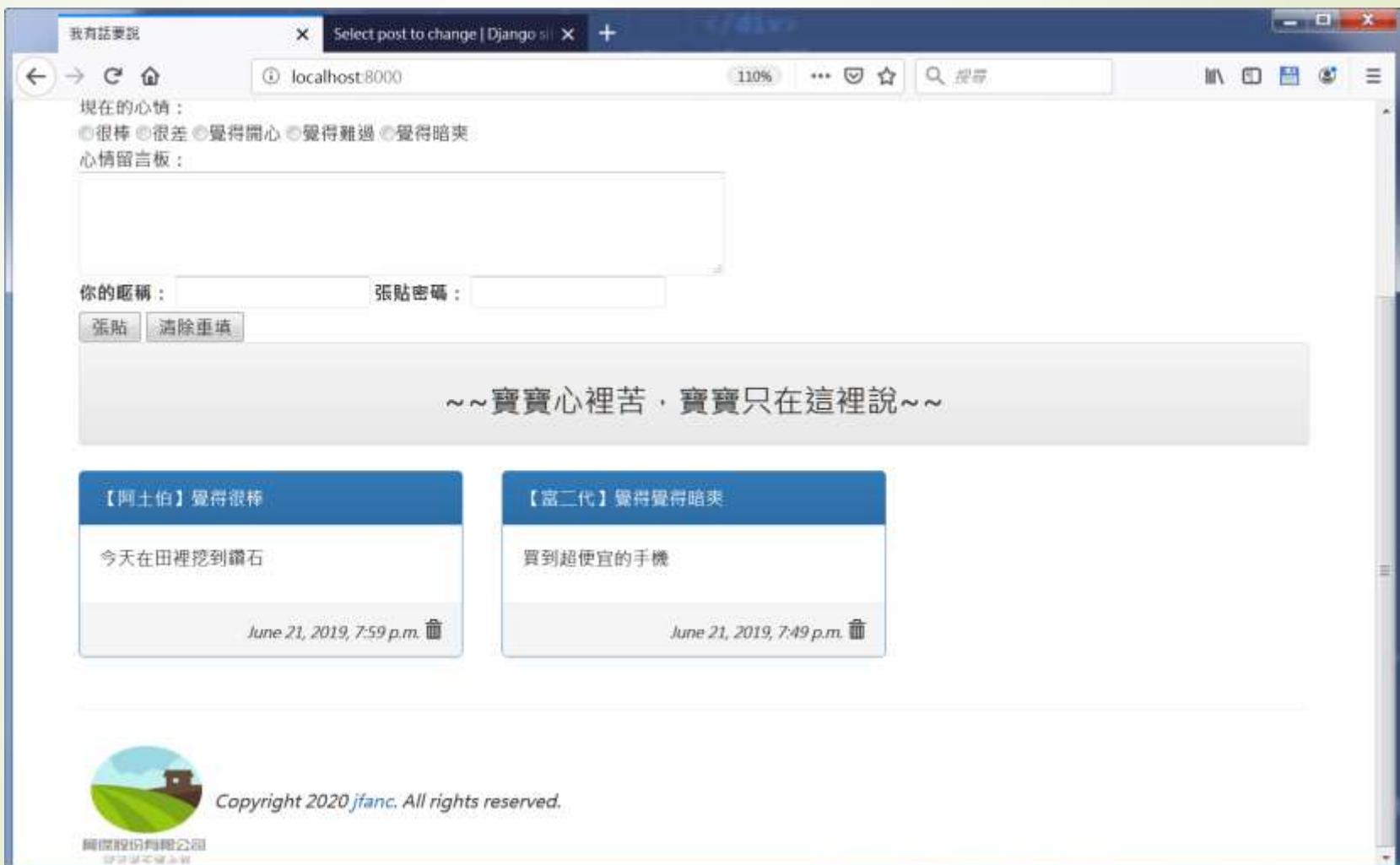

Copyright 2020 jfanc. All rights reserved.
蘭傑股份有限公司
台灣省新北市

加上刪除貼文的功能

- 範例中為每一篇貼文加上刪除密碼的欄位，也就是該貼文的作者在貼文的同時也必須設定一組密碼
- 先把密碼的欄位標題改成為張貼時以及刪除時都可以使用的密碼
- 要刪訊息時，只要在密碼欄填入資料，再按下該貼文的刪除符號（垃圾桶符號）

加上刪除貼文的功能

► 提供刪除貼文功能的網頁設計



加上刪除貼文的功能

- ▶ 在每一個貼文的右下角多了一個垃圾桶的符號，此符號使用以下的方式達成：
 - ▶
- ▶ 在onclick的地方加上了Javascript的函數呼叫指令，內容如下：
 - ▶ <script>
 - ▶ function go_delete(id){
 - ▶ var user_pass = document.getElementById('user_pass').value;
 - ▶ if (user_pass != "") {
 - ▶ var usr = '/' + id + '/' + user_pass;
 - ▶ window.location = usr;
 - ▶ }
 - ▶ else{
 - ▶ alert("請填刪除密碼");
 - ▶ }
 - ▶ }
 - ▶ </script>

加上刪除貼文的功能

► [urls.py](#)的對應樣式就要增加上此網址樣式，如下：

```
► urlpatterns = [  
    ►     path('admin/', admin.site.urls),  
    ►     path("", views.index),  
    ►     path('<int:pid>/<str:del_pass>', views.index),  
    ► ]
```

加上刪除貼文的功能

► 新版本的views.index如下所示：

```
► def index(request, pid=None, del_pass=None):  
►     posts = models.Post.objects.filter(enabled = True).order_by('-pub_time')[:30]  
►     moods = models.Mood.objects.all()  
►     try:  
►         user_id = request.GET['user_id']  
►         user_pass = request.GET['user_pass']  
►         user_post = request.GET['user_post']  
►         user_mood = request.GET['mood']  
►     except:  
►         user_id = None  
►         message = '如要張貼訊息，則每一個欄位都要填...'  
►
```

加上刪除貼文的功能

```
▶ if del_pass and pid:  
▶     try:  
▶         post = models.Post.objects.get(id=pid)  
▶     except:  
▶         post = None  
▶     if post:  
▶         if post.del_pass == del_pass:  
▶             post.delete()  
▶             message = "資料刪除成功"  
▶         else:  
▶             message = "密碼錯誤"  
▶     elif user_id != None:  
▶         mood = models.Mood.objects.get(status=user_mood)  
▶         post = models.Post.objects.create(mood=mood, nickname=user_id,  
del_pass=user_pass, message=user_post)  
▶         post.save()  
▶         message='成功儲存！請記得你的編輯密碼{}!，訊息需經審查後才會顯示。  
.format(user_pass)  
▶ return render(request, 'index.html', locals())
```

密碼錯誤

現在的心情：

很棒 很差 覺得開心 覺得難過 覺得暗爽

心情留言板：

你的暱稱：

張貼密碼：

張貼

清除重填

~~寶寶心裡苦，寶寶只在這裡說~~

【sdf】覺得覺得難過

asdf

June 21, 2019, 8:04 p.m.

【阿土伯】覺得很棒

今天在田裡挖到鑽石

June 21, 2019, 7:59 p.m.

【富二代】覺得覺得暗爽

買到超便宜的手機

June 21, 2019, 7:49 p.m.

我有話要說 X Select post to change | Django site index.html X

localhost:8000/6/sdf 110% 搜尋

NTU手機賣場 Home 後台管理

資料刪除成功

現在的心情：
◎很棒 ◎很差 ◎覺得開心 ◎覺得難過 ◎覺得暗爽

心情留言板：

你的暱稱： 張貼密碼：

張貼 清除重填

~~寶寶心裡苦，寶寶只在這裡說~~

【阿土伯】覺得很棒
今天在田裡挖到鑽石
June 21, 2019, 7:59 p.m. 垃圾箱

【富二代】覺得覺得暗爽
買到超便宜的手機
June 21, 2019, 7:49 p.m. 垃圾箱


Copyright 2020 jfanc. All rights reserved.
阿土伯股份有限公司

使用POST傳遞表單資料

- ▶ 本節中持續修改
- ▶ 新增一個網頁 [\(list\)](#) 只顯示所有的訊息
- ▶ 另外一個網頁 [\(post\)](#) 只可以用來張貼訊息
- ▶ 在 [urls.py](#) 中新增兩個網址樣式
- ▶ [views.py](#) 增加相對應的兩個處理函數
 - ▶ [views.listing\(\)](#)
 - ▶ [views.posting\(\)](#)

[在Internet Explorer 中URL 最大長度為2,083 個字元 - Microsoft Support](#)

使用POST傳遞表單資料

▶ 如下所示：

```
▶ urlpatterns = [  
    ▶   path('admin/', admin.site.urls),  
    ▶   path("/", views.index),  
    ▶   path('<int:pid>/<str:del_pass>', views.index),  
    ▶   path('list/', views.listing),  
    ▶   path('post/', views.posting),  
    ▶ ]
```

使用POST傳遞表單資料

- 在 `header.html` 中增加這兩個網址當做是選單之一
 - `留言列表`
 - `留言版`
- `view.listing()` 因為只負責顯示訊息，程式內容可大幅簡化如下：
 - `def listing(request):`
 - `posts =`
`models.Post.objects.filter(enabled=True).order_by('-pub_time')[:150]`
 - `moods = models.Mood.objects.all()`
 - `return render(request, 'listing.html', locals())`

使用POST傳遞表單資料

► 相對應的listing.html則做了一些顯示上的調整，如下所示：

```
► <!-- listing.html (HTML_Form02) -->
► {% extends "base.html" %}
► {% block title %}我有話要說{% endblock %}
► {% block content %}
► <div class='container'>
►   <div class='row'>
►     <div class='col-md-12'>
►       <div class='panel panel-default'>
►         <div class='panel-heading' align=center>
►           <h3>~~寶寶心裡苦，寶寶只在這裡說~~</h3>
►         </div>
►       </div>
►     </div>
►   </div>
► </div>
```

使用POST傳遞表單資料

```
▶ <div class="row">
▶   {%- for p in posts %}
▶     <div class="col-sm-12 col-md-6">
▶       <div class='panel' {%-cycle "panel-primary" "panel-info" "panel-warning" "panel-success" %}>
▶         <div class='panel-heading'>
▶           <table width='100%'>
▶             <tr>
▶               <td>
▶                 【{{ p.nickname }}】覺得{{ p.mood }}
▶               </td>
▶               <td align=right>
▶                 <i><small>{{ p.pub_time }}</small></i>
▶               </td>
▶             </tr>
▶           </table>
▶         </div>
▶       </div>
```

使用POST傳遞表單資料

```
▶ <div class='panel-body'>{{ p.message | linebreaks }}</div>
  ▶ </div>
  ▶ </div>
  ▶ {% endfor %}
  ▶ </div>
  ▶ </div>
  ▶ {% endblock %}
```

使用POST傳遞表單資料

► 調整後的外觀，如下圖：



使用POST傳遞表單資料

► 建立一個只做貼文功能的posting.html，程式片段如下所示：

```
► <!-- posting.html (HTML_Form02) -->
► {% extends "base.html" %}
► {% block title %}我有話要說{% endblock %}
► {% block content %}
► <div class='container'>
► {% if message %}
►   <div class='alert alert-warning'>{{ message }}</div>
► {% endif %}
```

使用POST傳遞表單資料

```
▶ <form name='my form' action='.' method='POST'>
▶     現在的心情 : <br/>
▶     {% for m in moods %}
▶         <input type='radio' name='mood' value='{{ m.status }}'>{{ m.status }}
▶     {% endfor %}
▶     <br/>
▶     心情留言板 : <br/>
▶     <textarea name='user_post' rows=3 cols=70></textarea><br/>
▶     <label for='user_id'>你的暱稱 : </label>
▶     <input id='user_id' type='text' name='user_id'>
▶     <label for='user_pass'>張貼/刪除密碼 : </label>
▶     <input id='user_pass' type='password' name='user_pass'><br/>
▶     <input type='submit' value='張貼'>
▶     <input type='reset' value='清除重填'>
▶     </form>
▶     </div>
▶     {% endblock %}
```

使用POST傳遞表單資料

► `views.posting` 函數如下所示：

```
► def posting(request):
    ►     template = get_template('posting.html')
    ►     moods = models.Mood.objects.all()
    ►     message = '如要張貼訊息，則每一個欄位都要填...'
    ►     html = template.render(locals())
    ►
    ► return HttpResponse(html)
    ►
    ► Django 1.8 之後不能使用以上程式碼
```

使用POST傳遞表單資料

► **header.html** 之中增加

- 留言列表
- 留言版

使用POST傳遞表單資料

► posting.html的網頁執行結果

我有話要說

localhost:8000/post/

NTU手機賣場 Home 留言列表 留言版 後台管理

如要張貼訊息，則每一個欄位都要填...

現在的心情：

很棒 很差 覺得開心 覺得難過 覺得暗爽

心情留言板：

你的暱稱： 張貼/刪除密碼：

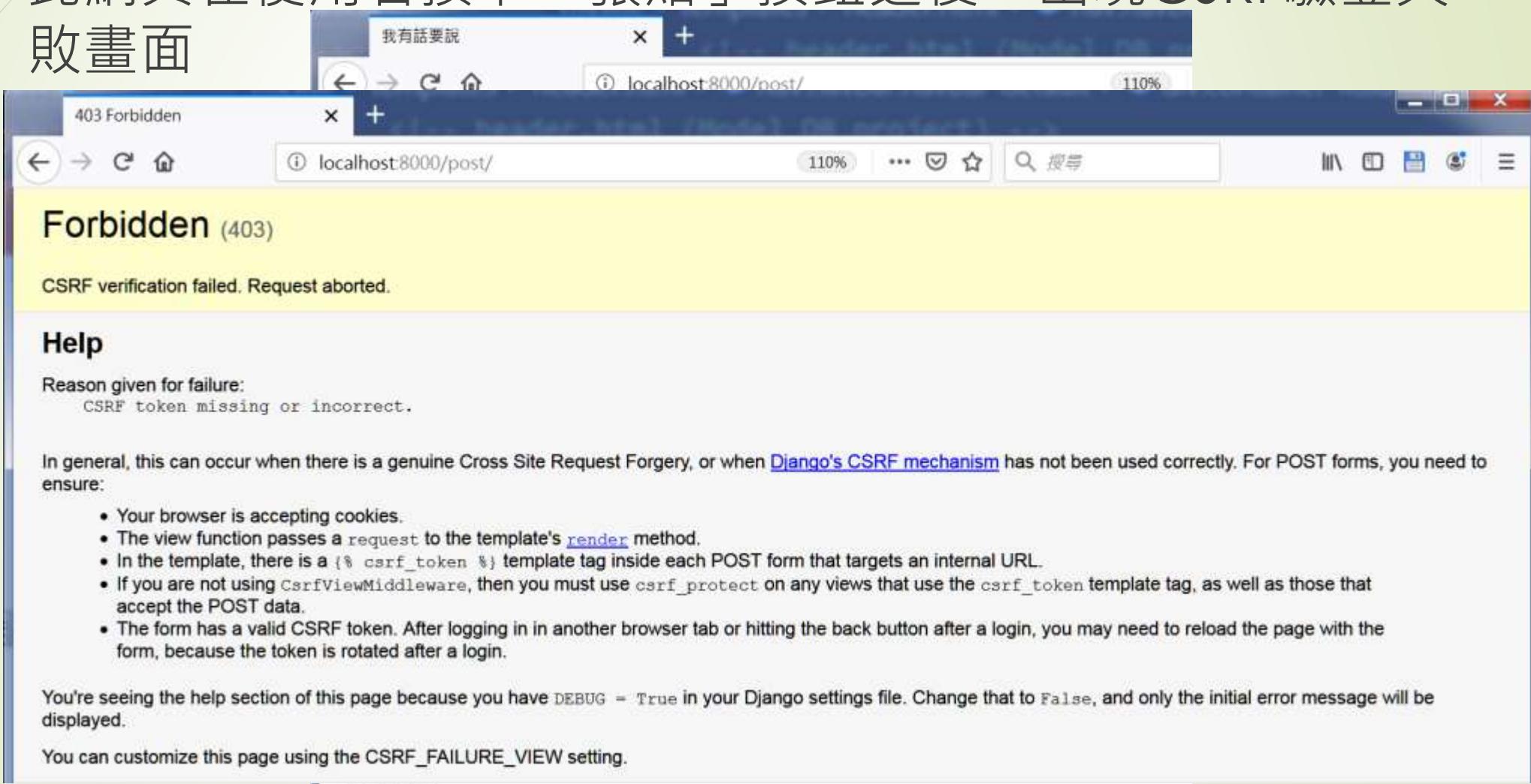
張貼 清除重填

Copyright 2020 jfanc. All rights reserved.

聯經股份有限公司
好读书读好书

使用POST傳遞表單資料

►此網頁在使用者按下「張貼」按鈕之後，出現CSRF驗登失敗畫面



使用POST傳遞表單資料

- ▶ 這是一個Django為了防止網站CSRF(Cross-site request forgery)攻擊的機制
- ▶ 確保駭客無法偽裝是已被驗證過的瀏覽器而盜取資料
- ▶ 啟用這個功能（預設是啟用的）的設定在settings.py中的'django.middleware.csrf.CsrfViewMiddleware'（在MIDDLEWARE_CLASSES的設定區塊）
- ▶ 配合此安全機制，我們還必須在posting.html中的`<form>`標籤底下加上如下所示的識別字：
 - ▶ `{% csrf_token %}`

```
<form name='my form' action='.' method='POST'>
    {% csrf_token %}
    現在的心情：<br/>
```

```
<form name='my form' action='.' method='POST'>
    <input type="hidden" name="csrfmiddlewaretoken" value="9mmq5ZWVdGNW9aV1BMmlu8yt44mdVwfk9sdhDokBz7yNTgXPHKopYIqCuyEkynx6">
    現在的心情：<br/>

    <input type='radio' name='mood' value='很棒'>很棒
    <input type='radio' name='mood' value='很差'>很差
    <input type='radio' name='mood' value='覺得開心'>覺得開心
    <input type='radio' name='mood' value='覺得難過'>覺得難過
    <input type='radio' name='mood' value='覺得暗爽'>覺得暗爽
```

使用POST傳遞表單資料

► 註：在Django 1.8以前，`csrf`機制常使用`RequestContext`方法來處理渲染，然而官方文件1.8（請參照網址一）說明，因為multiple template engines相容性問題，所以`RequestContext`等幾項方法在Django 1.10（請參照網址二）後將被移除，模板內容傳遞改使用dictionary方式。

► 網址一：

► <https://docs.djangoproject.com/en/1.10/releases/1.8/#current-app-argument-of-template-related-apis>

► 網址二：

► <https://docs.djangoproject.com/en/1.11/releases/1.11/#django-template-backends-django-template-render-prohibits-non-dict-context>

```
...
def posting(request):
    template = get_template('posting.html')
    moods = models.Mood.objects.all()
    message = '如要張貼訊息，則每一個欄位都要填...'
    html = template.render(locals())
    return HttpResponse(html)
...
def posting(request):
    moods = models.Mood.objects.all()
    message = '如要張貼訊息，則每一個欄位都要填...'
    return render(request, 'posting.html', locals())
```

- ▶ def posting(request):
- ▶ moods = models.Mood.objects.all()
- ▶ message = '如要張貼訊息，則每一個欄位都要填...'
- ▶ return render(request, 'posting.html', locals())

可以送了，但貼文還收不到

The screenshot shows a web browser window with the URL <http://localhost:8000/post/>. The page title is "我有話要說". The navigation bar includes links for "NTU手機賣場", "Home", "留言列表", "留言版", and "後台管理". A yellow message box contains the text: "如要張貼訊息，則每一個欄位都要填...". Below it, there's a section for心情选择 with radio buttons: "很棒", "很差", "覺得開心", "覺得難過", and "覺得暗爽". A text area labeled "心情留言板：" is empty. At the bottom, there are input fields for "你的暱稱：" and "張貼/刪除密碼：" with corresponding buttons "張貼" and "清除重填". The footer features a logo of a house on a hill and the text "Copyright 2020 jfanc. All rights reserved.".

小練習-新增post方法

- ▶ 從 views.index複製收get的方法
- ▶ 把get改post之後
- ▶ 用一樣的方式寫入資料庫

```
user_id = request.POST['user_id']
user_pass = request.POST['user_pass']
user_post = request.POST['user_post']
user_mood = request.POST['mood']
```

基礎表單類別的應用

- ▶ 除了用手工的方式把表單使用HTML標籤一個一個刻上去之外，其實就像是Models一樣
- ▶ Django也提供好用的表單類別，讓我們以程式的方式來產生表單以便應用於Template模板檔案中，而且功能也非常多

結合表單和資料庫

- ▶ Django本身提供現成可使用的表單類別Form和ModelForm，可以使用物件導向的方式，直接利用程式碼產生需要的表單內容
- ▶ 假設要為網站設計一個連絡表單，讓網友可以透過這個表單寄個電子郵件給網站管理員，如果使用表單類別的方式，需先定義一個自訂的表單類別
 - ▶ 在mysite目錄之下另外建立一個forms.py檔案
 - ▶ 引入django的forms模組
 - ▶ 繼承forms.Form建立一個網站要使用的自訂表單類別
 - ▶ 在此類別中指名要使用的欄位內容

結合表單和資料庫

- 典型的連絡表單看起來會像是下面這個樣子 (`forms.py`) :

```
#_*_ encoding: utf-8 *_*
from django import forms

class ContactForm(forms.Form):
    CITY = [
        ['TP', 'Taipei'],
        ['TY', 'Taoyuang'],
        ['TC', 'Taichung'],
        ['TN', 'Tainan'],
        ['KS', 'Kaohsiung'],
        ['NA', 'Others'],
    ]
    user_name = forms.CharField(label='您的姓名', max_length=50, initial='小智')
    user_city = forms.ChoiceField(label='居住城市', choices=CITY)
    user_school = forms.BooleanField(label='是否在學', required=False)
    user_email = forms.EmailField(label='電子郵件')
    user_message = forms.CharField(label='您的意見', widget=forms.Textarea)
```

結合表單和資料庫 - 內建表單

- ▶ `forms.py` 中示範了幾個欄位的設定範例，就像是之前在設定 Models 類別一樣，要建立一個 **類別**（在此例為 `ContactForm`），然後分別設定其中的 **欄位名稱**（在此例分別是 `user_name`, `user_city`, `user_school`, `user_email`, `user_message`）
- ▶ 種常見的欄位格式設定方法，摘要說明如下表：

欄位格式名稱	用法	說明
<code>CharField</code>	<code>CharField(label=' 您 的 姓 名 ', max_length=50, initial='李大仁')</code>	<code>label</code> 為本欄位的標籤（以下皆同）， <code>max_length</code> 設定長度為 50， <code>initial</code> 為欄位中的預設值，在本例為'李大仁'

結合表單和資料庫

ChoiceField	ChoiceField(label='居住城市', choices=CITY)	設定下拉式選單（即<select>標籤），後面需以choices參數指定一個二維串列，如程式中的CITY
BooleanField	BooleanField(label='是否在學', required=False)	布林值的欄位，即checkbox標籤，required設定為False則此checkbox在輸入時也可以不用勾選
EmailField	EmailField(label='電子郵件')	具email驗證功能的欄位
CharField forms.Textarea	+ CharField(label='您的意見', widget=forms.Textarea)	在CharField中以widget=forms.Textarea來擴增成為大量文字輸入欄位，亦即<textarea>標籤。

結合表單和資料庫

- 一旦建立了此格式的類別，即可在views.contact處理函數中產生實例
- 假設我們已在urls.py中建立了/contact的網址樣式
- 對應到views.contact的內容如下：
 - from mysite import models, forms
 - def contact(request):
 - form = forms.ContactForm()
 - return render(request, 'contact.html', locals())
- header.html之中增加
 - 連絡我

結合表單和資料庫

► [contact.html](#)中的內容如下：

- ▶ <!-- contact.html (HTML_Form02) -->
- ▶ {% extends "base.html" %}
- ▶ {% block title %}連絡管理員{% endblock %}
- ▶ {% block content %}
- ▶ <div class='container'>
- ▶ <div class='panel panel-primary'>
- ▶ <div class='panel-heading'>
- ▶ <form name='my form' action='.' method='POST'>
- ▶ {% csrf_token %}
- ▶ <h3>寫信給管理員</h3>
- ▶ </div>
- ▶ <div class='panel-body'>
- ▶ {{ form.as_p }}



```
    ─▶ </div>
    ─▶   <div class='panel-footer'>
    ─▶     <input type='submit' value='送出'>
    ─▶   </form>
    ─▶ </div>
    ─▶ </div>
    ─▶ </div>
    ─▶ {%
    ─▶   endblock %}
```

結合表單和資料庫

► contact.html的執行畫面

The screenshot shows a web page titled '寫信給管理員' (Write to Admin). The top navigation bar includes links for 'NTU手機賣場', 'Home', '留言列表', '留言版', '連絡我', and '後台管理'. The main form has the following fields:

- 您的姓名: 李大仁
- 居住城市: Taipei
- 是否在學:
- 電子郵件:
- 您的意見:
- 送出

At the bottom left is a logo for '網儒股份有限公司' (JiFanc Co., Ltd.) featuring a stylized building on a hill. The copyright notice reads: 'Copyright 2020 JiFanc. All rights reserved.'

資料接收與欄位的驗證方法

- ▶ 在網頁上按下「送出」按鈕之後網站並不會有其它的反應
- ▶ 如何接收表單上的資料呢？
 - ▶ 先使用if判斷傳進來的內容是否為POST
 - ▶ 如果是，那麼表示剛剛連進來的request是因為按下表單中的submit按鈕來的
 - ▶ 程式如下所示：
 - ▶ def contact(request):
 - ▶ if request.method == 'POST':
 - ▶ form = forms.ContactForm(request.POST)
 - ▶ if form.is_valid():
 - ▶ message = "感謝您的來信。"
 - ▶ else:
 - ▶ message = "請檢查您輸入的資訊是否正確！"
 - ▶ else:
 - ▶ form = forms.ContactForm()
 - ▶ return render(request, 'contact.html', locals())

- ▶ <https://docs.djangoproject.com/zh-hans/2.2/ref/forms/validation/>

資料接收與欄位的驗證方法

► 在contact.html的前面多加了以下的內容：

- { % if message %}
- <div class='alert alert-warning'>
- {{ message }}
- </div>
- { % endif %}

The screenshot shows a contact form titled "寫信給管理員". The form fields include "您的姓名: 李大仁", "居住城市: Taipei", "是否在學: ", "電子郵件: abc@gmail.com", and a redacted message field with the word "asdf" underlined in red.

感謝您的來信。

寫信給管理員

您的姓名: 李大仁

居住城市: Taipei

是否在學:

電子郵件: abc@gmail.com

asdf

資料接收與欄位的驗證方法

► 表單類別的自動欄位檢查功能

A screenshot of a web browser window titled "localhost:8000/contact/". The page has a blue header bar with the title "寫信給管理員". Below it is a form with several fields:

- 姓名: 李大仁
- 居住城市: Taoyuang
- 是否在學:
- 電子郵件: (empty)
- 意見: (empty)

Validation errors are shown as red arrows pointing to the following fields:

- The "居住城市" field has an error message: "This field is required."
- The "電子郵件" field has an error message: "This field is required."

A screenshot of a contact form titled "寫信給管理員". The form fields include:

- 姓名: 李大仁
- 居住城市: Taipei
- 是否在學:
- 電子郵件: (empty)
- 意見: (empty)

The "電子郵件" field is highlighted with a red border and has an error message: "請填寫此欄位。".

A screenshot of a contact form titled "寫信給管理員". The form fields include:

- 姓名: 李大仁
- 居住城市: Taipei
- 是否在學:
- 電子郵件: abc
- 意見: (empty)

The "電子郵件" field is highlighted with a red border and has an error message: "請輸入一個電子郵件地址。".

← → C ① localhost:8000/contact/

NTU手機賣場 Home 留言列表 留言版 連絡我 後台管理

請檢查您輸入的資訊是否正確！

寫信給管理員

您的姓名: 小智

居住城市: Taipei ▼

是否在學:

- Enter a valid email address.

電子郵件: abc@asdf

abc

您的意見:

送出

資料接收與欄位的驗證方法

► 表單類別自動會幫我們處理這些細節

The screenshot shows a contact form titled "寫信給管理員" (Write to Admin) on a browser window. The URL in the address bar is "localhost:8000/contact/". The form fields include "您的姓名" (Name) with value "李大仁", "居住城市" (City) with value "Taoyuang", "是否在學" (Is in school) with checked radio button, and "電子郵件" (Email) with value "kkk". A red arrow points to a validation message for the email field: "請在電子郵件地址中包含「@」。'kkk' 未包含「@」。" (Please ensure the email address contains '@'. 'kkk' does not contain '@').

資料接收與欄位的驗證方法

► 在使用`form.is_valid()`檢查了表單的正確性之後，只要使用每一個表單原本在定義時使用的識別字，就可以取出其中的資料，方法如下：

```
► form = forms.ContactForm(request.POST)
► if form.is_valid():
►     message = "感謝您的來信。"
►     user_name = form.cleaned_data['user_name']
►     user_city = form.cleaned_data['user_city']
►     user_school = form.cleaned_data['user_school']
►     user_email = form.cleaned_data['user_email']
►     user_message = form.cleaned_data['user_message']
► else:
►     message = "請檢查您輸入的資訊是否正確！"
```

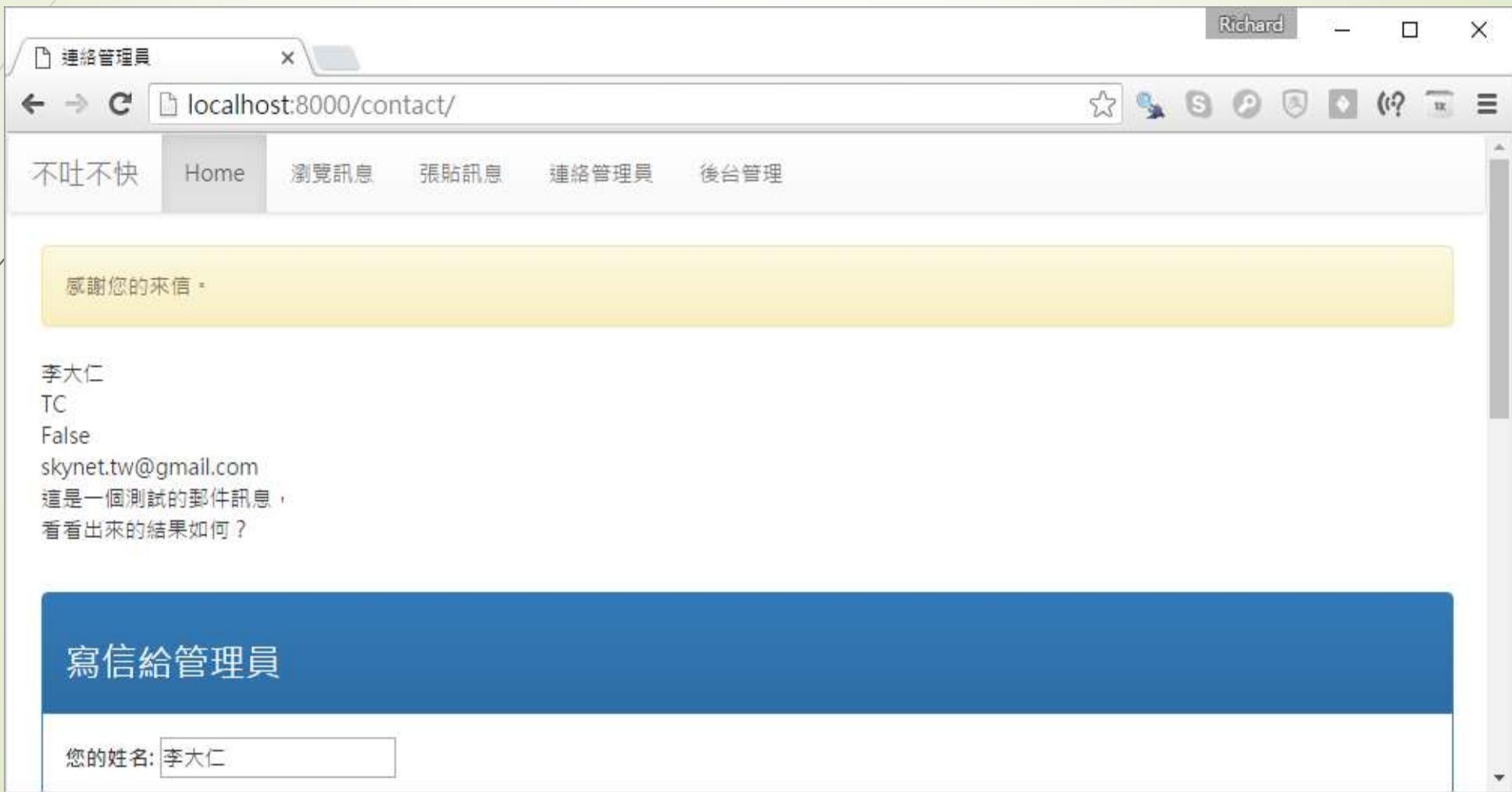
資料接收與欄位的驗證方法

► 拿到的資料直接顯示在網頁上的方法如下面這段程式碼：

```
►  {%- if message %}  
    <div class='alert alert-warning'>  
        {{ message }}  
    </div>  
    {{ user_name }}<br/>  
    {{ user_city }}<br/>  
    {{ user_school }}<br/>  
    {{ user_email }}<br/>  
    {{ user_message | linebreaks }}<br/>  
  {%- endif %}
```

資料接收與欄位的驗證方法

► 顯示表單中的資料



利用第三方服務郵寄電子郵件

► 本小節就以Mailgun為例，說明如何在Django網站中使用Mailgun的郵件寄送服務

- 前往<http://mailgun.com>註冊一個自己的帳號
- 啟用之後即會為你設定一個寄送的帳號名稱（看起來像是 sandboxecfb1a3816e94a06...8f139.mailgun.org這樣網址）
- 在自己的電腦中使用pip安裝django-mailgun模組，如下所示：
 - (ENV) D:\myDjango\HTML_Form><pip install django-mailgun>

Try Now

[cURL](#)[Python](#)[Java](#)[Go](#)[Ruby](#)[PH](#)

```
1 def send_simple_message():
2     return requests.post(
3         "https://api.mailgun.net/v3/sandboxe7019cfdc0f73d5c53.mailgun.org/messages",
4         auth=("api", "748082ad4c9dbebd8d2d0c7bba33b8l"),
5         data={"from": "Mailgun Sandbox <postmaster@sandboxe7019cfdc0f73d5c53.mailgun.org>",
6               "to": "張傑 <shiunyi71@gmail.com>",
7               "subject": "Hello 張傑",
8               "text": "Congratulations 張傑, you just sent an email with Mailgun! You are truly awesome!"})
9
10 # You can see a record of this email in your logs: https://app.mailgun.com/app/logs.
11
12 # You can send up to 300 emails/day from this sandbox server.
13 # Next, you should add your own domain so you can send 10000 emails/month for free.
```

利用第三方服務郵寄電子郵件

► 到 `settings.py` 中做好設定才能夠啟用此功能，在 `settings.py` 中的任何一處加入以下的 3 個常數即可：

- `EMAIL_BACKEND = 'django_mailgun.MailgunBackend'`
- `MAILGUN_ACCESS_KEY = 'key-55.....27e7'`
- `MAILGUN_SERVER_NAME = 'xxx.mailgun.org'`

► `MAILGUN_ACCESS_SKY` 以及 `MAILGUN_SERVER_NAME` 的內容可以到 mailgun 的帳號設定中，找到 Domain 的頁面，如下圖：

利用第三方服務郵寄電子郵件

► Mailgun的Domain設定頁面

The screenshot shows the Mailgun dashboard with a sidebar on the left and a main content area on the right.

Sidebar:

- @mailgun
- Dashboard
- Sending
- Domains
- Overview
- Logs
- Analytics
- Templates
- Suppressions
- Webhooks

Main Content Area:

Dashboard

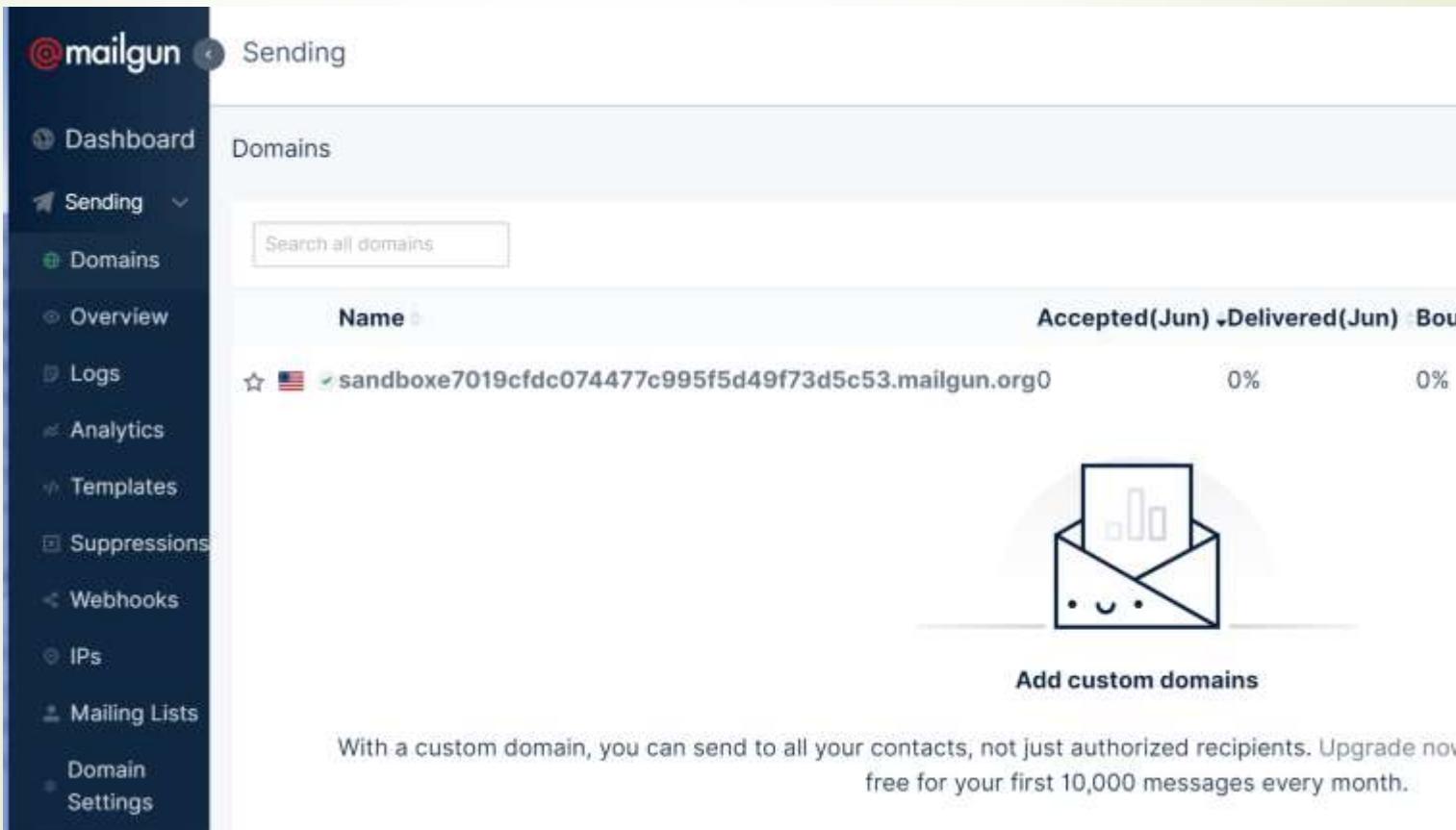
- 1 **Activate your account**
Check your inbox for a "Please verify your Mailgun account" email, click the link, and you're done!
- 2 **Add & verify a domain**
With your domain set up, you can send more emails, and the emails you send will come from you.
- 3 **Start sending**
To SMTP or API? That is the question. This guide [can help you decide.](#)
- 4 **Get better email delivery**
Enhance your reputation by sending from a dedicated IP and use Mailgun's email validation service to find real email addresses from real people on your list.

Buttons at the bottom:

- Try Now
- cURL
- Python

利用第三方服務郵寄電子郵件

- ▶ 如果已有一個網域名稱可以直接使用此網域，如果沒有新增自訂網域的朋友，請點擊那個sandbox開頭的預設網域



How would you like to send your emails from sandboxe7019cfdc074477c995f5d49f73d5c53.mailgun.org?

API

The most flexible, most popular way to send email.

Use languages like Ruby, Python, PHP, C# and more.

Select

SMTP

The easiest way to send email.

Grab your SMTP credentials and plug them into your app's settings.

Select

Sandbox domains are restricted to authorized recipients only.

Which language do you want to use?

cURL

Python

Java

Go

Ruby

PHP

C#

Node.js

API key: 748082ad40[REDACTED]72d0e9

API base URL: <https://api.mailgun.net/v3/sandboxe7019cfdc074477c995f5d49f73d5c53.mailgun.org>

Here's the basic code you need. Plug in your API info from above and modify the from address, to address, and other content to give Mailgun a good old test drive.

利用第三方服務郵寄電子郵件

- ▶ 在views.py中引入以下的模組：
 - ▶ from django.core.mail import EmailMessage
- ▶ views.contact中改寫如下：
 - ▶ def contact(request):
 - ▶ if request.method == 'POST':
 - ▶ form = forms.ContactForm(request.POST)
 - ▶ if form.is_valid():
 - ▶ message = "感謝您的來信，我們會儘速處理您的寶貴意見。"
 - ▶ user_name = form.cleaned_data['user_name']
 - ▶ user_city = form.cleaned_data['user_city']
 - ▶ user_school = form.cleaned_data['user_school']
 - ▶ user_email = form.cleaned_data['user_email']
 - ▶ user_message = form.cleaned_data['user_message']
 - ▶
 - ▶

利用第三方服務郵寄電子郵件

```
► mail_body = u""  
► 網友姓名 : {}  
► 居住城市 : {}  
► 是否在學 : {}  
► 反應意見 : 如下  
► "{}".format(user_name, user_city, user_school, user_message)  
  
► email = EmailMessage( '來自【NTU亂亂賣】網站的網友意見',  
►                         mail_body,  
►                         user_email,  
►                         [' my.email@gmail.com ']) #管理員(你自己的)email  
► email.send()  
► else:  
►     message = "請檢查您輸入的資訊是否正確！"  
► else:  
►     form = forms.ContactForm()  
► return render(request, 'contact.html', locals())
```

利用第三方服務郵寄電子郵件

► 張貼訊息的功能

連接管理員 × @ sandboxe7019cfcd074477c995 × Screenshot-20190907-134146 × +

localhost:8000/contact/

寫信給管理員

感謝您的來信，我們會儘速處理您的寶貴意見。

您的姓名: 李小狼

居住城市: Tainan

是否在學:

電子郵件: xiomin@ntu.edu.tw

天氣真好

您的意見: _____

送出

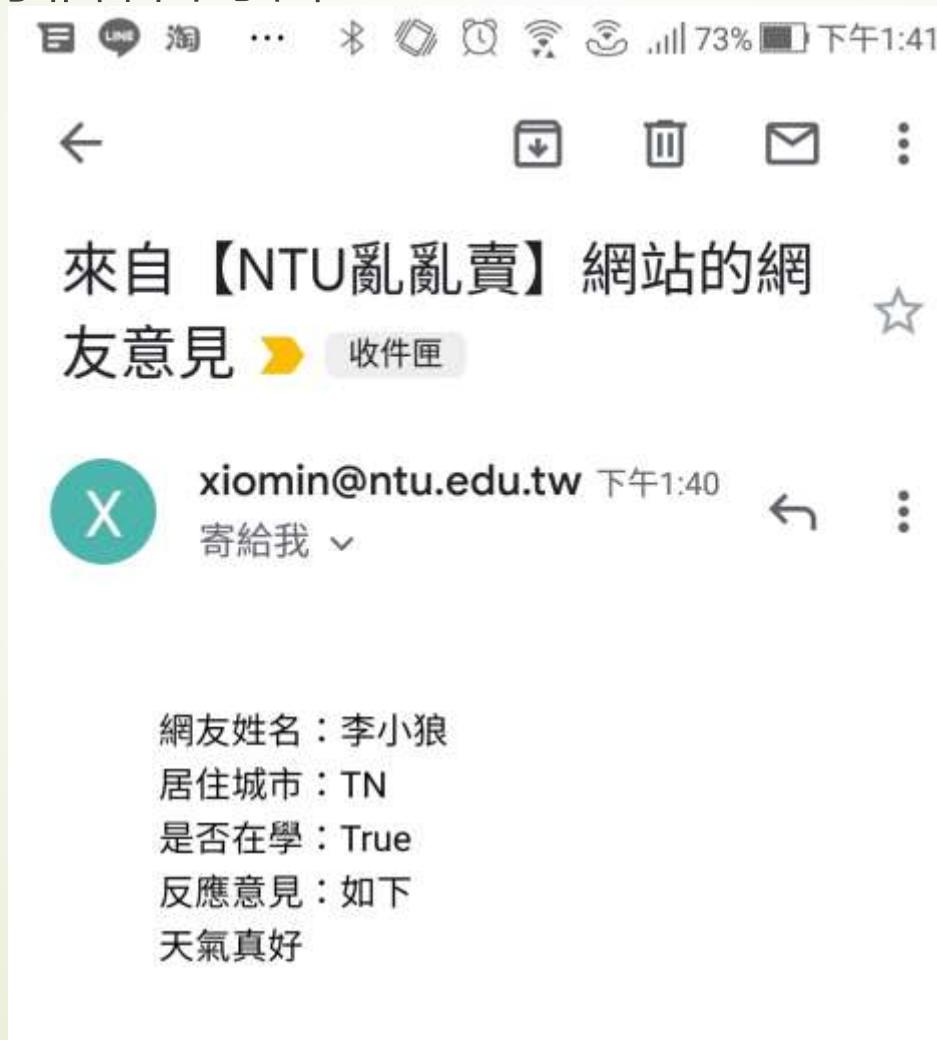
利用第三方服務郵寄電子郵件

► 寄信成功的網頁畫面



利用第三方服務郵寄電子郵件

► 收到的信件內容



但是以上第三方email已經不支援
(更新後又支援了！)

- ▶ 可以改用**Anymail: Django email integration for transactional ESPs**
- ▶ <https://github.com/anymail/django-anymail>

```
from django.core.mail import send_mail
from django.core.mail import EmailMultiAlternatives
from anonymail.message import attach_inline_image_file

def contact(request):
    if request.method == 'POST':
        form = forms.ContactForm(request.POST)
        if form.is_valid():
            message = "感謝您的來信，我們會儘速處理您的寶貴意見。
"
            user_name = form.cleaned_data['user_name']
            user_city = form.cleaned_data['user_city']
            user_school = form.cleaned_data['user_school']
            user_email = form.cleaned_data['user_email']
            user_message = form.cleaned_data['user_message']

            mail_body = u'''
網友姓名：{}
居住城市：{}
是否在學：{}
反應意見：如下
{}'''.format(user_name, user_city, user_school,
user_message)
```

```
msg = EmailMultiAlternatives(  
    subject="來自【NTU亂亂賣】網站的網友意見",  
    body=mail_body,  
    from_email=user_email,  
    to=["shiunyi71@gmail.com"],  
    reply_to=["Helpdesk <support@example.com>"])  
  
msg.send()  
else:  
    message = "請檢查您輸入的資訊是否正確！"  
else:  
    form = forms.ContactForm()  
return render(request, 'contact.html', locals())
```

寫信給管理員

您的姓名: 李奧納多一卡皮箱

居住城市: Taipei ▼

是否在學:

電子郵件: pikachu@ntu.edu.tw

今天天氣真好！

小智在哪裡？

您的意見:

送出

感謝您的來信，我們會儘速處理您的寶貴意見。

李奧納多一卡皮箱

TP

True

pikachu@ntu.edu.tw

今天天氣真好！

小智在哪裡？



Copyright 2020 jfanc. All rights reserved.

阿傑股份有限公司



撰寫



第 1 個，共 28,645 個



注

收件匣

46

已加星號

已延後

重要郵件

寄件備份

草稿

33

所有郵件

類別

台大郵件

3

有用資訊

資訊系統訓練... 8

CS+X 15

有趣的信

Comment o... 2

來自【NTU亂亂賣】網站的網友意見 ➔ 收件匣 ×



pikachu@ntu.edu.tw 透過 sandbox.mgsend.net

寄給我 ▾

下午11:45 (1 分鐘前)



網友姓名：李奧納多一卡皮箱

居住城市：TP

是否在學：True

反應意見：如下

今天天氣真好！

小智在哪裡？



回覆



轉寄

其它發送email的方式

- ▶ <https://docs.djangoproject.com/en/2.2/topics/email/>
- ▶ <https://code.ziqiangxuetang.com/django/django-send-email.html>
- ▶ <https://juejin.im/entry/5ad9ddaa6fb9a07acd4d4a2e>
- ▶ <https://simpleisbetterthancomplex.com/tutorial/2016/06/13/how-to-send-email.html#basic-django-email-functions>
- ▶ <https://stackoverflow.com/questions/6367014/how-to-send-email-via-django>

模型表單類別ModelForm的應用

- ▶ 如果網站是屬於資料庫為主的網站，也已經建立了資料模型 Model的話，那麼不用說，可以直接整合Model和Form的表單就是最佳的選擇。
- ▶ 直接在表單中套用在Model中定義好的欄位在表單上，還可以決定要保留那些欄位，非常實用
- ▶ <https://docs.djangoproject.com/zh-hans/2.2/topics/forms/modelforms/#creating-forms-from-models>

ModelForm的使用

- ▶ 為了方便比較，另外創建了一個post2db的網址（修改urls.py）
 - ▶ path('post2db/', views.post2db),
- ▶ 以及post2db.html和views.post2db函數
- ▶ 在這裡面使用ModelForm來產生表單
 - ▶ 在forms.py中加入PostForm這個類別，這次要繼承自ModelForm，如下：
 - ▶ from . import models
- ▶ class PostForm(forms.ModelForm):
 - ▶ class Meta:
 - ▶ model = models.Post
 - ▶ fields = ['mood', 'nickname', 'message', 'del_pass']

ModelForm的使用

```
► def __init__(self, *args, **kwargs):  
    ►     super(PostForm, self).__init__(*args, **kwargs)  
    ►     self.fields['mood'].label = '現在心情'  
    ►     self.fields['nickname'].label = '你的暱稱'  
    ►     self.fields['message'].label = '心情留言'  
    ►     self.fields['del_pass'].label = '設定密碼'
```

ModelForm的使用

► 在views.post2db的程式內容中，先設計為如下（暫時還不處理回應的表單內容，單純就顯示的部份）：

```
► def post2db(request):
    ►     post_form = forms.PostForm()
    ►     moods = models.Mood.objects.all()
    ►     message = '如要張貼訊息，則每一個欄位都要填...'
    ►     return render(request, 'post2db.html', locals())
```

ModelForm的使用

► [post2db.html](#)，程式如下所示：

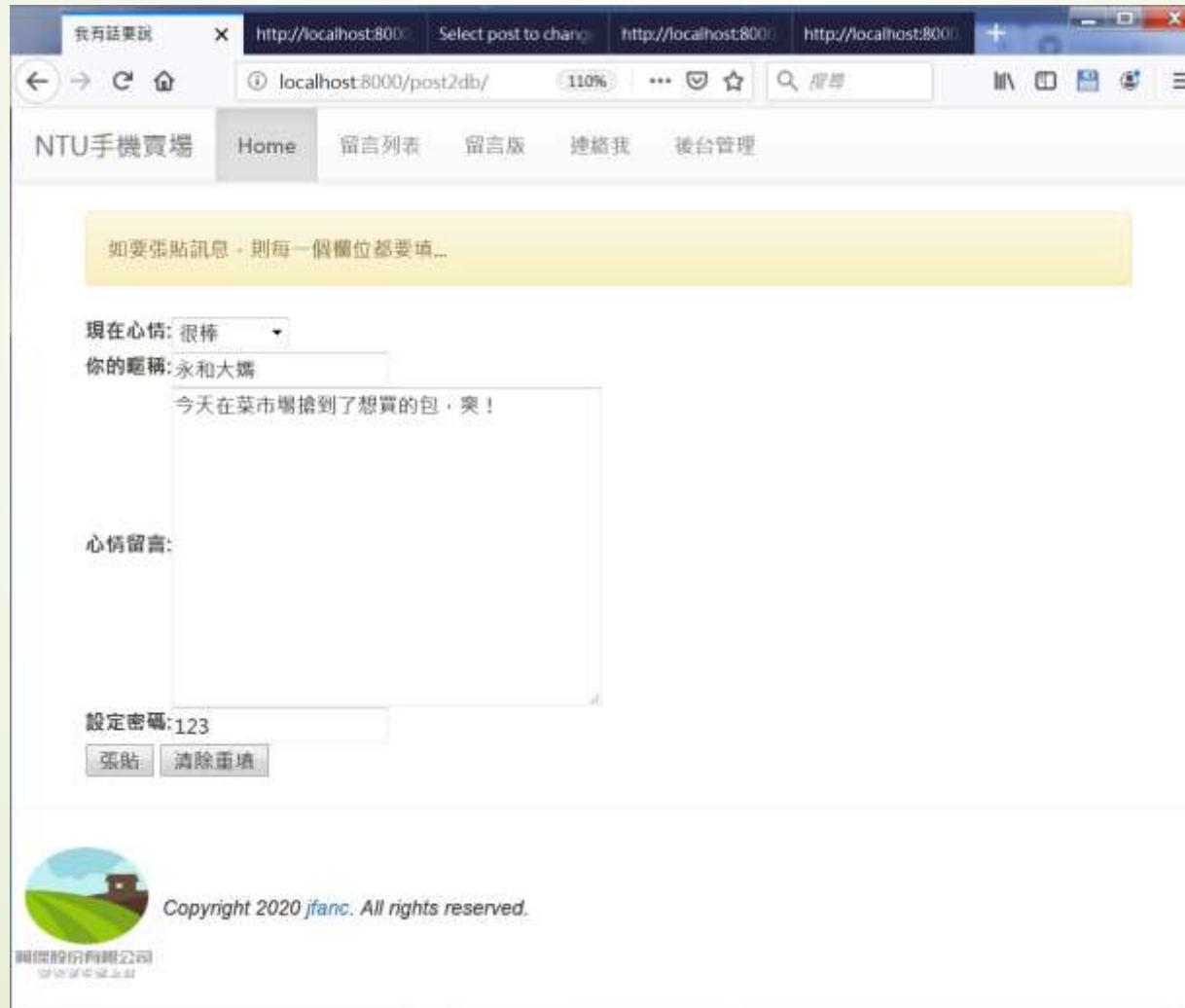
```
► <!-- post2db.html (HTML_Form02) -->
► {% extends "base.html" %}
► {% block title %}我有話要說{% endblock %}
► {% block content %}
► <div class='container'>
► {% if message %}
►   <div class='alert alert-
warning'>{{ message }}</div>
► {% endif %}
```

ModelForm的使用

```
▶<form name='my form' action='.' method='POST'>
  ▶  {% csrf_token %}
  ▶  <table>
  ▶    {{ post_form.as_table }}
  ▶  </table>
  ▶  <input type='submit' value='張貼'>
  ▶  <input type='reset' value='清除重填'>
  ▶</form>
  ▶</div>
  ▶{% endblock %}
```

ModelForm的使用

► 使用ModelForm表單所顯示的網頁內容



透過ModelForm產生的表單儲存資料

► 要如何儲存資料呢？

► 先檢查輸入的部份，再檢查正確性，最後再以`save()`函數儲存即可，如下所示：

```
► def post2db(request):
    ►     if request.method == 'POST':
        ►         post_form = forms.PostForm(request.POST)
        ►         if post_form.is_valid():
            ►             message = "您的訊息已儲存，要等管理者啟用後才看得到喔。"
            ►             post_form.save()
        ►     else:
            ►         message = '如要張貼訊息，則每一個欄位都要填...'
        ► else:
            ►     post_form = forms.PostForm()
            ►     message = '如要張貼訊息，則每一個欄位都要填...'
    ► return render(request, 'post2db.html', locals())
```

透過ModelForm產生的表單儲存資料

► 儲存訊息之後的網頁畫面



透過ModelForm產生的表單儲存資料

- ▶ 這個程式碼會在張貼完之後仍然停留在原來的網頁以及表單資料，我們也可以使用HttpResponseRedirect重新導向另外一個網頁，常見的情形是導向首頁或是訊息瀏覽頁面，只要在post_form.save()的下一行加入如下這一行敘述即可：
 - ▶ return HttpResponseRedirect('/list/')
- ▶ 在程式檔的最前面，也要引入這個模組才行，如下所示：
 - ▶ from django.http import HttpResponseRedirect

我有話要說 view-source:localhost:8000/list Select post to ch sandboxe7019c ibb Screenshot-2019-09-01 at 10.56.11.11

Action: ----- Go 0 of 7 selected

NTU手機賣場 NICKNAME 小智 qq

如要張貼訊 現在心情: 非常開心
你的暱稱: 小智
終於成功抓到比比鳥了！

心情留言:
設定密碼: 123
張貼 清除重填

NTU手機賣場 Home 後台管理 留言列表 留言版 留言版2db 連絡我

~~寶寶心裡苦，寶寶只在這裡說~~

【小智】覺得非常開心 Sept. 7, 2019, 6:17 a.m.
終於成功抓到比比鳥了！

【abc】覺得非常開心 Sept. 1, 2019, 7:01 a.m.
終於可以po文了，而且可以顯示囉！

【阿土伯】覺得覺得非常暗爽 Sept. 1, 2019, 6:01 a.m.
今天種田的時候
挖到金塊
好開心
但又不能講

【qq】覺得傷心 Sept. 1, 2019, 5:58 a.m.
執行範例一直失敗

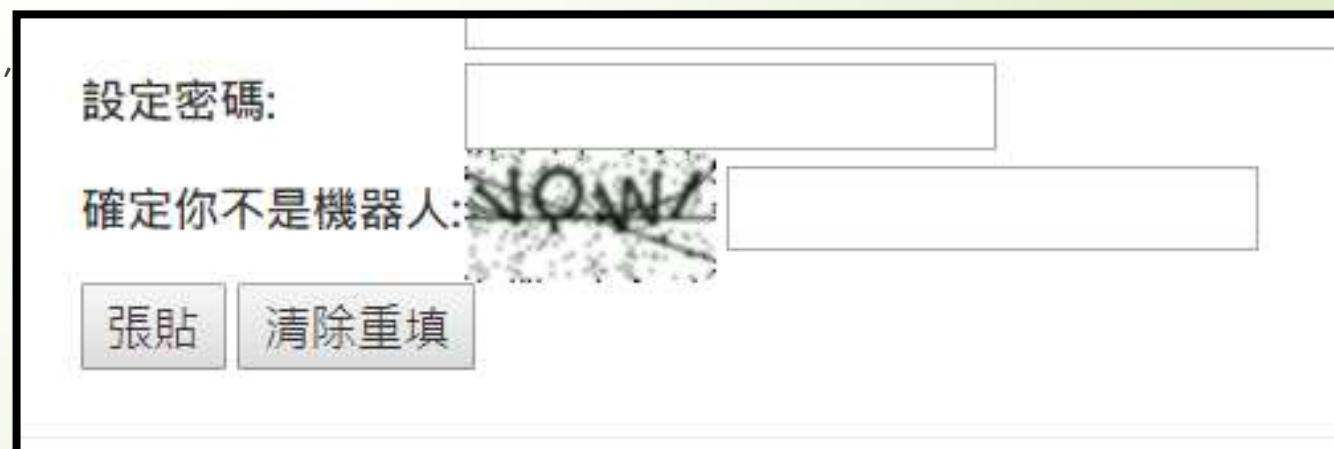
【999】覺得很開心 Sept. 1, 2019, 5:49 a.m.
因為範例可以正常執行了

Copyright 2020 jfanc. All rights reserved.

天天玩樂股份有限公司 每天都可以上癮

為表單加上防機器人驗證機制

- ▶ 安裝django-simple-captcha模組，如下所示：
 - ▶ (VENV) D:\myDjango\HTML_Form02>pip install django-simple-captcha
- ▶ 在settings.py中把'captcha'加到INSTALLED_APP的區塊中，如下所示：
 - ▶ INSTALLED_APPS = (
 - ▶ 'django.contrib.admin',
 - ▶ 'django.contrib.auth',
 - ▶ 'django.contrib.contenttypes',
 - ▶ 'django.contrib.sessions',
 - ▶ 'django.contrib.messages',
 - ▶ 'django.contrib.staticfiles',
 - ▶ 'mysite',
 - ▶ '**captcha**',
 - ▶)



為表單加上防機器人驗證機制

- ▶ 執行資料庫的migrate動作如下：
 - ▶ (VENV) D:\myDjango\HTML_Form02>`python manage.py migrate`
- ▶ 在`urls.py`加上這個模組專用的網址對應：
 - ▶ `url(r'^captcha/', include('captcha.urls')),`
 - ▶ `from django.conf.urls import include`
 - ▶ `path('captcha/', include('captcha.urls')),`

為表單加上防機器人驗證機制

► 確定Pillow是否已安裝在系統中，可以使用pip list檢查一下，如果找不到Pillow版本的話，設法把Pillow安裝到系統中：

- (ENV) D:\myDjango\HTML_Form>pip list
 - dds (1.0)
 - Django (1.8.14)
 - django-mailgun (0.9.1)
 - django-simple-captcha (0.5.1)
 - MySQL-python (1.2.5)
 - Pillow (3.3.0)
 - pip (8.1.2)
 - requests (2.10.0)
 - setuptools (24.0.3)
 - six (1.10.0)
 - wheel (0.29.0)

為表單加上防機器人驗證機制

► 以上都沒有問題的話，就可以直接在forms.py中加上CaptchaField了，如下所示：

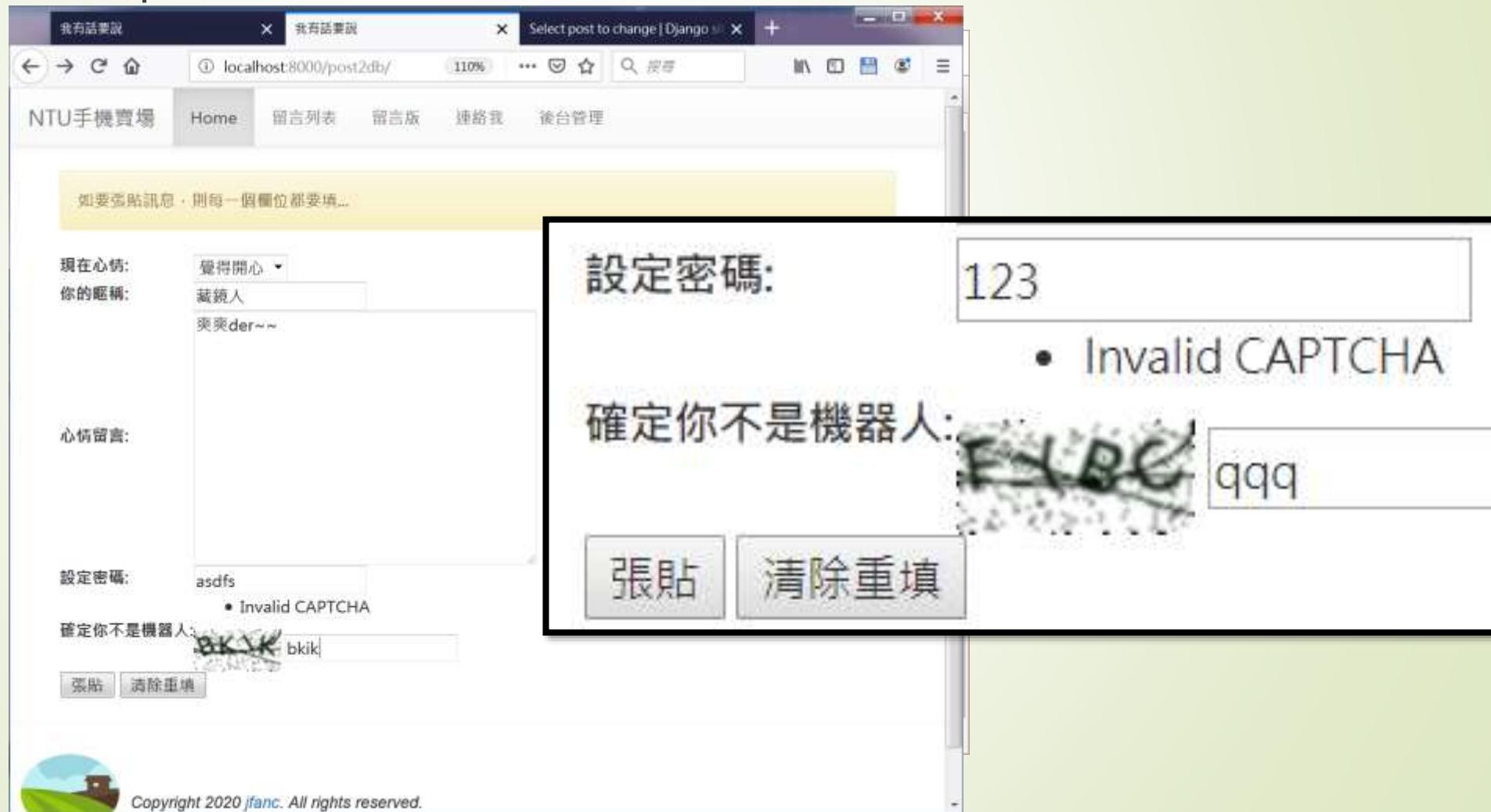
```
► from captcha.fields import CaptchaField  
► class PostForm(forms.ModelForm):  
    ►     captcha = CaptchaField()  
    ►     class Meta:  
        ►         model = models.Post  
        ►         fields = ['mood', 'nickname', 'message', 'del_pass']  
    ►
```

為表單加上防機器人驗證機制

```
▶ def __init__(self, *args, **kwargs):  
    super(PostForm, self).__init__(*args, **kwargs)  
    self.fields['mood'].label = '現在心情'  
    self.fields['nickname'].label = '你的暱稱'  
    self.fields['message'].label = '心情留言'  
    self.fields['del_pass'].label = '設定密碼'  
    self.fields['captcha'].label = '確定你不是機器人'
```

為表單加上防機器人驗證機制

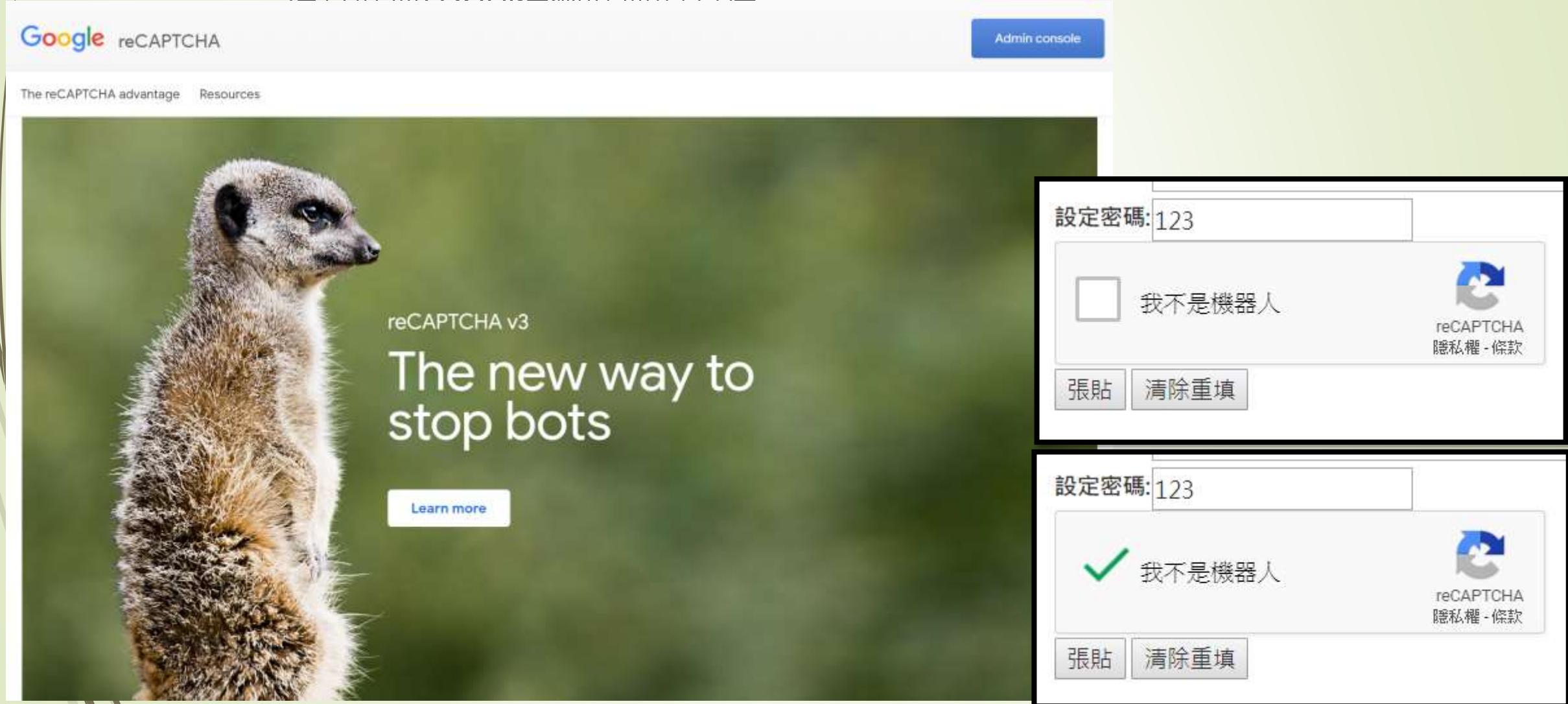
► 加上Captcha圖形驗證的表單畫面



為表單加上防機器人驗證機制

128

- Google也有提供reCAPTCHA機器人驗證的API，首先我們要到reCAPTCHA admin(網址：<https://www.google.com/recaptcha/intro/android.html>)進行註冊I abel填寫註冊的名稱



► 接著點選reCAPTCHA V2

► 最後輸入你網站IP，直接使用127.0.0.1本地IP就可以了，如下圖：
► 註冊Google reCAPTCHA





13

網域 (i)

+ 127.0.0.1

擁有者

shiunyi71@gmail.com (你)

輸入電子郵件地址

 接受《reCAPTCHA 服務條款》

一旦存取或使用 reCAPTCHA API，即表示您同意遵守《Google API 使用條款》、《Google 使用條款》，以及下列附加條款。請詳閱並瞭解所有適用條款與政策後再存取 API。

reCAPTCHA 服務條款 ▾

 傳送通知給擁有者 (i)[取消](#)[提交](#)



13

在你的網站中加進 reCAPTCHA

「HTML_Form04」已完成註冊。

在向使用者顯示的 HTML 程式碼中使用這串網站金鑰。

[進一步瞭解用戶端整合](#)

複製網站金鑰

6Ld5L6sUAAAAAAE

NzuMgJt-c4k

用這串密鑰來建立網站和 reCAPTCHA 之間的通訊。

[進一步瞭解伺服器端整合](#)

複製密鑰

6Ld5L6sUAAAAAAE

APHpep0G

[前往設定](#)

[前往 ANALYTICS \(分析\)](#)



為表單加上防機器人驗證機制

- 按下註冊後，會有一個Site key與Secret key
 - Site key
 - 用在「客戶端」HTML，提供 reCAPTCHA 驗證碼服務給使用者
 - Secret key
 - 用在「伺服器端」，網站與 Google 之間的通訊
- 把Secret key加到settings.py內，如下所示：
 - GOOGLE_RECAPTCHA_SECRET_KEY = '你的Secret key 放在這'

為表單加上防機器人驗證機制

► [post2db.html](#)表單內放置reCAPTCHA的小工具，用以做機器人驗證，如下所示：

- { % block content %}
- <div class='container'>
- { % if message %}
- <div class='alert alert-warning'>{{ message }}</div>
- { % endif %}

- <form name='my form' action='.' method='POST'>
- { % csrf_token %}

為表單加上防機器人驗證機制

```
► <table>          script中為Google reCAPTCHA提供用以驗證的API  
►   {{ post_form.as_table }}  
► </table>  
► <script src='https://www.google.com/recaptcha/api.js'></script>  
► <div class="g-recaptcha" data-sitekey="**"></div>  
► <input type='submit' value='張貼'>  
► <input type='reset' value='清除重填'>  
► </form>  
► </div>  
► {%- endblock %}
```

放置你的public_key

► 在views.py中post2db方法內加入驗證程式碼，顯示如下：

```
► import json  
► import urllib  
► from django.conf import settings  
► def post2db(request):  
►     if request.method == 'POST':  
►         post_form = forms.PostForm(request.POST)  
►         if post_form.is_valid():  
►             recaptcha_response = request.POST.get('g-recaptcha-response')  
►             url = 'https://www.google.com/recaptcha/api/siteverify'
```

為表單加上防機器人驗證機制

```
► values = {  
►     'secret': settings.GOOGLE_RECAPTCHA_SECRET_KEY,  
►     'response': recaptcha_response  
► }  
► data = urllib.parse.urlencode(values).encode()  
► req = urllib.request.Request(url, data=data)  
► response = urllib.request.urlopen(req)  
► result = json.loads(response.read().decode())  
► if result['success']:  
►     message = "您的訊息已儲存，要等管理者啟用後才看得到喔。"  
►     post_form.save()  
►     return HttpResponseRedirect('/list/')  
► else:  
►     message = "reCAPTCHA驗證失敗，請在確認."  
► else:  
►     message = '如要張貼訊息，則每一個欄位都要填...'  
► else:  
►     post_form = forms.PostForm()  
►     message = '如要張貼訊息，則每一個欄位都要填...'  
► return render(request, 'post2db.html', locals())
```

到 forms.py 把上一個captcha註解掉

```
class PostForm(forms.ModelForm):
    #captcha = CaptchaField()
    class Meta:
        model = models.Post
        fields = ['mood', 'nickname', 'message',
                  'del_pass']
    def __init__(self, *args, **kwargs):
        super(PostForm, self).__init__(*args, **kwargs)
        self.fields['mood'].label = '現在心情'
        self.fields['nickname'].label = '你的暱稱'
        self.fields['message'].label = '心情留言'
        self.fields['del_pass'].label = '設定密碼'
        #self.fields['captcha'].label = '確定你不是機器人'
```

► 加入Google reCAPTCHA

140



我有話要說 × +

localhost:8000/list/ 110% ⋮ 🔍 搜尋

NTU手機賣場 Home 留言列表 留言版 留言版2db 連絡我 後台管理

~~寶寶心裡苦，寶寶只在這裡說~~

【不願意透漏身份的人】覺得覺得開心 June 28, 2019, 3:29 p.m.

123愛台灣

【不願意透漏身份的人2】覺得很差 June 23, 2019, 12:38 a.m.

sdsdfdf

【孫小美】覺得覺得難過 June 22, 2019, 10:55 p.m.

我的鑽石掉了

【阿土伯】覺得很棒 June 21, 2019, 7:59 p.m.

今天在田裡挖到鑽石

【富二代】覺得覺得暗爽 June 21, 2019, 7:49 p.m.

回家作業

1. 請比較Form和ModelForm的差別。
2. 請完成以下功能的網站：
3. 如果想要讓使用者在張貼訊息時直接就可以啟用該訊息，不需要經由管理者的操作，需要改哪一個地方？
4. 請在張貼訊息的功能加上張貼密碼的驗證，通過密碼的驗證即可直接顯示該訊息。
5. 同上，如果有提供密碼，該訊息即自動張貼並啟用，如果沒有提供密碼，則改為需要管理者啟用才可顯示。

補充

- ▶ <https://docs.djangoproject.com/zh-hans/2.2/ref/forms/>
- ▶ <https://docs.djangoproject.com/zh-hans/2.2/topics/forms/>
- ▶ <https://docs.djangoproject.com/zh-hans/2.2/topics/forms/modelforms/>