

Crear un Template Tags en Django

14 de Febrero de 2017 a las 01:01

Las plantillas(template) en Django muestra los datos pasados por una vista, esos datos son presentados habitualmente mediante código HTML y una serie de etiquetas(tag), variables y filtros, mirar documentación Django [aquí](#). Las etiquetas permite aumentar la funcionalidad de las plantillas con código de python, por ejemplo permite usar condicionales y bucles.

Django proporciona dentro de su motor de plantillas una serie de etiquetas que podemos añadir, mas información [aquí](#), pero también se pueden crear etiquetas para nuestros proyectos, es lo que se denomina *template tags*.

Esta entrada no explicará los *template tags* en profundidad y los tipos que hay. Mi intención es mostrar un caso práctico de como utilizarlos. Se utilizará como ejemplo la creación de una sección de "Entradas Recientes" en un blog.

Se utilizará Django 1.10 y Python 3.5.2.

CASO PRÁCTICO

Tenemos un blog y queremos añadir en un panel lateral una sección de "*Entradas recientes*" donde se mostrara el título de los últimas 4 entradas añadidas al blog. Esta sección será dinámica porque deberá cambiar si se producen cualquier cambio en el blog podría afectar a la lista de *Entradas Recientes*.

Para crear esta sección se utilizará un *template tags* del tipo de *inclusión*, [inclusion tags](#), que permite insertar un trozo de código dentro de una plantilla.

Primero debemos realizar una consulta en la base de datos del *blog* y que devuelva el título de los 4 cuatros últimas entradas añadidas. En mi caso esta consulta seria.

```
'post_list': Entry.objects.all().order_by("-publish", "-created")[:4]
```

Esta consulta selecciona todos las entradas almacenadas en la tabla *Entry*, los ordena por entradas publicadas y fecha de creación("-publish","-created"), mostrando solo las 4 primeras ([:4]).

Ahora para crear un *template tags*, creamos en nuestro proyecto *Django* una carpeta "*templatetags*" y en esa carpeta creamos un fichero con una función donde insertaremos la consulta anterior y el *inclusion tags*

El archivo quedaría de la siguiente forma.

```
from django import template
register = template.Library()
```

Blog de J.A. Jimenez Toro, rooteando.com

Contenido esta bajo licencia [Creative Commons Reconocimiento No Comercial Sin Obra Derivada 3.0 Unported License](#)

```
@register.inclusion_tag('snippets/recent_post.html')
def render_recent_blogposts():
    return {
        'post_list': Entry.objects.all().order_by("-publish", "-created")[:4]
    }
```

Tenemos una función `render_recent_blogposts` donde encontramos la consulta que se ha mostrado anteriormente. Justo encima de la definición de la función tenemos el código para el *inclusion tags*.

```
@register.inclusion_tag('snippets/recent_post.html')
```

Registramos un *template tags* del tipo *inclusion* entre los paréntesis incluimos el código a insertar, en este caso el código está en un archivo y se especifica la ruta. Por último habrá que importar la librerías de *template* y crear una variable para registrar el *template tags*.

Ahora falta crear el archivo `recent_post.html`.

```
{% for post in post_list %}
<li><a href="{{ post.get_absolute_url }}"> {{ post.title }}</a></li>
{% endfor %}
```

Este archivo permite poner en la lista el título de la entrada y su enlace, si se pulsa en el título se mostrara la entrada en el blog.

Como resultado tendremos

Entradas recientes

- [Instalar Open Sans en un proyecto de Django y CKEditor](#)
- [Generar un PDF de un post en Django](#)
- [Arkos vs YunoHost](#)
- [Virtualenvwrapper, LiClipse, pip y algunas cosas mas](#)

Por último, recordad mi canal de *Telegram* [Un Día Una Aplicación](https://telegram.me/UnDiaUnaAplicacion) <https://telegram.me/UnDiaUnaAplicacion>, donde cada día se mostrara una aplicación, distribución, comando... de Linux, también su cuenta de Twitter asociada [@UnDiaUnaApp](https://twitter.com/UnDiaUnaApp).