

Python web con Django

Módulo 1



Vistas y direcciones de URL

Vistas y direcciones de URL

Cuando el servidor web recibe una petición, la pasa a un proyecto de Django para que devuelva una respuesta y pueda ser enviada al navegador. Cuando el proyecto recibe una petición (esto es, cuando un usuario ingresa a una dirección de URL cualquiera del sitio), se invoca una función definida en alguna de las aplicaciones del proyecto para que retorne una respuesta. Ahí es donde el desarrollador se hace cargo del control del sitio y puede ejecutar código de Python.

Entonces, lo primero que vamos a hacer es crear una función dentro de nuestro archivo de nombre ***myapp/views.py***

Borremos todo código anterior y escribamos lo siguiente:

```
from django.http import HttpResponse
def index ( request ):
    return HttpResponse( "¡Hola, mundo!" )
```



Aquí se debe importar la clase **HttpResponse** del módulo **django.http**. (Las clases son como funciones, pero la diferencia es que siempre retornan un valor y siguen una convención de nombramiento diferente: la primera letra de cada palabra en mayúscula y sin guiones bajos). A través de esta clase es posible crear la respuesta (response) que se desea devolver al navegador web. El argumento que se le pasa a **HttpResponse** es una cadena con el contenido de esa respuesta. A las funciones que retornan una *instancia* (i. e. el valor de retorno) de **HttpResponse** se las conoce como *vistas*, de ahí proviene el nombre del archivo **views.py**

Ahora bien, es necesario indicarle a Django a qué dirección de URL debe responder esta función **index()** . Para ello, se crea el archivo **myapps/urls.py** con el siguiente código:

```
from django.urls import path
from . import views
urlpatterns = [
    path( "" , views.index, name = "index" )
]
```

El código anterior puede parecer un poco extraño, pero sencillamente es la estructura que Django eligió para el archivo `urls.py` de cada aplicación, donde se realizan las asociaciones entre direcciones de URL y vistas. Las asociaciones se incluyen como elementos de la lista `urlpatterns`.



Para cada uno de ellos, usando la función `django.urls.path()` se indica (siguiendo el orden de los argumentos):

1. La dirección de URL a la que debe responder una vista (en este caso, la cadena vacía indica la página de inicio, que según se mencionó más arriba se le daba el nombre de `index`).
2. Una función que recibe como argumento una petición (`request`).
3. Una cadena que identifique a la función anterior (veremos más adelante qué utilidad se le da a esto).

Nota: en el desarrollo de aplicaciones de Django, es común utilizar rutas relativas al importar un módulo. Recordemos que nuestra aplicación **myapp** es un paquete de Python, que Django va a importar para invocar las vistas (o sea, llamar las funciones) definidas en el módulo **myapp.views**.

Por defecto, el directorio de trabajo de una aplicación de Django es la ruta del proyecto que la está utilizando (en este caso, la carpeta **myproject**, donde se encuentra el archivo **manage.py**). Cuando en el archivo **myapp/urls.py** se necesita importar el módulo **myapp/views.py**, no se hace simplemente `import views`, porque esto buscaría el archivo **views.py** en la carpeta **myproject**.

En su lugar: **`from . import views`**

En este contexto el punto indica que el módulo **views** está junto al archivo desde donde se lo está importando; o sea, en este caso, que **views.py** y **urls.py** están en la misma ubicación (**myproject/myapp**). Una forma alternativa de importar las vistas sería la siguiente:

`from myapp import views`

o bien

`import myapp.views as views`

Esta no es una sintaxis poco habitual, pero tiene un problema: si luego se cambia el nombre del paquete **myapp**, se deben alterar todas las líneas donde se lo haya importado.



Hasta aquí, se han configurado las direcciones de URL de *la aplicación myapp*. Ahora, debemos agregar esta configuración particular a la configuración general del proyecto (recordar que un proyecto puede tener varias aplicaciones).

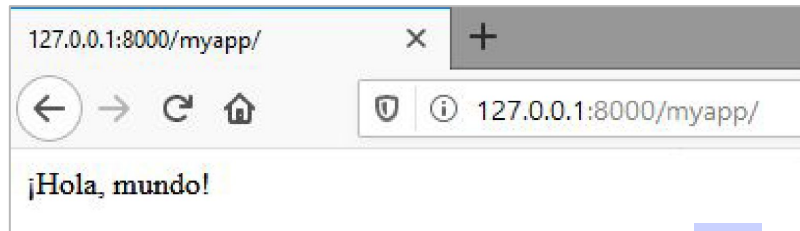
Para ello, se abre el archivo **myproject/urls.py** y se reemplaza su contenido por el siguiente:

```
from django.urls import path, include
urlpatterns = [
    path( "myapp/" , include( "myapp.urls" ))
]
```

Con este código se le indica a Django: en el proyecto, se desea que todas las direcciones de URL que comienzan con **myapp/** (por ejemplo: <http://127.0.0.1:8000/myapp/> o <http://dominio.com/myapp/>) sean derivadas a la aplicación **myapp**.



Lo que siga a esa dirección de URL (por ejemplo, **/contacto**) será administrado por la configuración de las direcciones de URL de esa aplicación, esto es, por el archivo **myapp/urls.py** que se creó anteriormente. Como allí se definió que la vista `index()` debía responder a una cadena vacía, al visitar <http://127.0.0.1:8000/myapp/> se verá el contenido de la respuesta que se retornó en esa función.



D



Se hará este procedimiento una vez más para esclarecerlo un poco. Se agrega la siguiente vista a la aplicación en el archivo `myapp/views.py`:

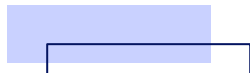
```
def acerca_de ( request ):  
    return HttpResponse( "¡Curso de Python y Django!" )  
Ahora se la asocia con una dirección de URL; puede ser, /acerca-de. Para eso, en  
myapp/urls.py se agrega un elemento a la lista urlpatterns:  
urlpatterns = [  
    path( "" , views.index, name = "index" ),  
    path( "acerca-de" , views.acerca_de, name = "acerca_de" )  
]
```



Por último, se visita <http://127.0.0.1:8000/myapp/acerca-de> y efectivamente se observa:



Nótese que las direcciones de URL definidas para una aplicación son relativas a la configuración que hayan recibido en el proyecto en el que está incluida.



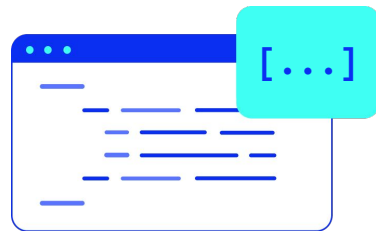
El prefijo **myapp/** fue definido por el proyecto en el archivo **myproject/urls.py**, mientras que **acerca-de** fue definido en la aplicación, de ahí que la dirección de URL final sea:

<http://127.0.0.1:8000/myapp/acerca-de>

Si un proyecto contiene únicamente una aplicación, se puede quitar el prefijo definiendo una cadena vacía en **myproject/urls.py**:

```
urlpatterns = [  
    path( "", include( "myapp.urls" ))  
]
```

Si este fuera el caso, ahora la dirección de URL para la vista **myapp.views. acerca_de()** sería limpiamente <http://127.0.0.1:8000/acerca-de>.



**¡Sigamos
trabajando!**

