

HINTS

ENRUTAMIENTO

Django maneja una solicitud enrutando la ruta URL entrante a una función de vista. La función de visualización es responsable de devolver una respuesta al cliente que realiza la solicitud. Las diferentes URL son manejadas generalmente por diferentes funciones de vista. Para dirigir la solicitud a una función de vista específica, Django analiza la configuración de su URL (o **URLconf** para abreviar). La plantilla de proyecto predeterminada define el URLconf en: **<myproject>/urls.py**.

Su **URLconf** debe ser un módulo de Python que define un atributo denominado **urlpatterns**, que es una lista de **django.conf.urls.url()**. Cada instancia de **url()** debe, como mínimo, definir una expresión regular (una expresión regular) para que coincida con la URL y un objetivo, que es una función de vista o un URLconf diferente. Si un patrón de URL apunta a una función de vista, es una buena idea darle un nombre para hacer referencia fácilmente al patrón más adelante.

Revisemos un ejemplo básico:

```
1 # In <myproject>/urls.py
2
3 from django.conf.urls import url
4
5 from myapp.views import home, about, blog_detail
6
7 urlpatterns = [
8     url(r'^$', home, name='home'),
9     url(r'^about/$', about, name='about'),
10    url(r'^blog/(?P<id>\d+)/$', blog_detail, name='blog-detail'),
11 ]
```

Este **URLconf** define tres patrones de URL, todos dirigidos a una vista: **home**, **about** y **blog-detail**.

```
1 url(r'^$', home, name='home'),
```

La expresión regular contiene un anclaje de inicio **'^** , seguido inmediatamente por un anclaje de extremo **'\$** . Este patrón coincidirá con las solicitudes en las que la ruta de la URL sea una cadena vacía, y las **myapp.views** a la vista de **home** definida en **myapp.views**.

```
1 url(r'^about/$', about, name='about'),
```

Esta expresión regular contiene un anclaje de inicio, seguido por la cadena literal `about/`, y el anclaje final. Esto coincidirá con la URL `/about/`, y lo dirigirá a la vista `about`. Dado que cada URL no vacía comienza con un `/`, Django corta convenientemente la primera barra.

```
1 url(r'^blog/(?P<id>\d+)/$', blog_detail, name='blog-detail'),
```

Aquí define el ancla de inicio y la cadena literal `blog/`, como el patrón anterior. La siguiente parte, `(?P<id>\d+)`, se llama un grupo de captura. Éste toma una parte de la cadena, y Django la pasa como un argumento a la función de vista.

La sintaxis de un grupo de captura es `(?P<name>pattern)`. `name` define el nombre del grupo, que también es el nombre que Django usa para pasar el argumento a la vista. El patrón define qué caracteres coinciden con el grupo.

En este caso, el nombre es `id`, por lo que la función `blog_detail` debe aceptar un parámetro llamado `id`. El patrón es `\d+`. `\d` significa que el patrón sólo coincide con caracteres numéricos. Y `+` significa que el patrón debe coincidir con uno, o más caracteres.

Algunos patrones comunes:

- `\d+` Uno o más caracteres numéricos.
- `[\w-]+` Uno o más caracteres alfanuméricos, guiones bajos o guiones.
- `[0-9]{4}` año (largo) Cuatro números, de cero a nueve.
- `[0-9]{2}` año (corto) mes día del mes. Dos números, de cero a nueve.
- `[^/]+` Segmento de trayectoria. Cualquier cosa excepto una barra.

El grupo de captura en el patrón de `blog-detail` está seguido por un literal `/`, y el ancla final.

Las URL válidas incluyen:

```
1 /blog/1/ # passes id='1'
2 /blog/42/ # passes id='42'
```

Las URL inválidas son, por ejemplo:

```
1 /blog/a/ # 'a' does not match '\d'
2 /blog// # no characters in the capturing group does not match '+'
```

Django procesa cada patrón de URL en el mismo orden en que se definen en `urlpatterns`. Esto es importante si varios patrones pueden coincidir con la misma URL. Por ejemplo:

```
1 urlpatterns = [
2     url(r'blog/(?P<slug>[\w-]+)/$', blog_detail, name='blog-detail'),
3     url(r'blog/overview/$', blog_overview, name='blog-overview'),
4 ]
```

En el `URLconf` anterior, el segundo patrón no es accesible. El patrón coincidiría con la URL `/blog/overview/`, pero en lugar de llamar a la vista `blog_overview`, la URL primero coincidirá con el patrón de `blog-detail`, y llamará a la vista `blog_detail` con un argumento `slug='overview'`.

Para asegurarse de que la URL `/blog/overview/` se enrute a la vista `blog_overview`, el patrón se debe colocar sobre el patrón del `blog-detail`:

```
1 urlpatterns = [
2     url(r'blog/overview/$', blog_overview, name='blog-overview'),
3     url(r'blog/(?P<slug>[\w-]+)/$', blog_detail, name='blog-detail'),
4 ]
```