

Archivos estáticos

Antonio Espín Herranz

Configuración

- Una parte importante del desarrollo Web son los archivos estáticos.
- Para Django los archivos estáticos son: imágenes, css, javascript, etc.
- Estos recursos se alojan en una carpeta dentro de la aplicación, por convención esta carpeta se llama **static**, y dentro de esta podemos tener un contenido organizado en carpetas.
- Django proporciona el módulo: **django.contrib.staticfiles** (se añade por defecto al crear el proyecto en el fichero *settings.py*)

Configuración

- Comprobar en el fichero de configuración: **settings.py** que hay una entrada a **django.contrib.staticfiles** en la lista de la variable:

```
INSTALLED_APPS = [  
    ...  
    'django.contrib.staticfiles',  
    ...  
]
```

- Cuando creamos el proyecto de Django con el comando, **dentro del fichero settings.py** crea una **variable** de configuración que apunta a la ruta de la carpeta de contenido estático de la aplicación:

```
STATIC_URL = '/static/'
```

- Django no crea la carpeta, hay que crearla de forma manual.

Configuración

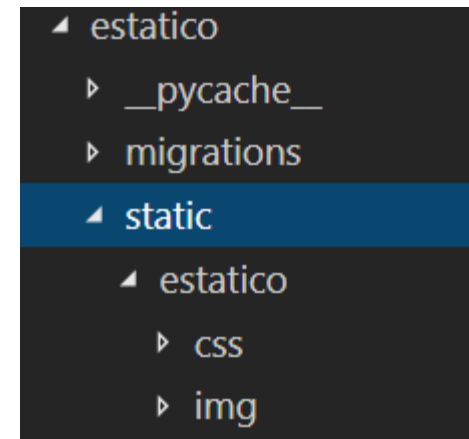
- El contenido estático se utilizará dentro de una plantilla, por ejemplo, para cargar una hoja estilos CSS:

```
<head>
    {% load static %}
    <link rel="stylesheet" href="{% static 'css/estilos.css' %}">
</head>
```

- Para cargar una imagen / logo / etc.

```
{% load static %}


```



Dentro de la carpeta **static** si tenemos varias subcarpetas (**una por aplicación**) se indicará En la ruta del fichero CSS se precede del nombre de la aplicación.
Por ejemplo: href="{% static '**estatico**/css/estilos.css' %}">

Configuración

- Django aconseja establecer las siguientes rutas:
`my_app/static/my_app/css/estilos.css`
`my_app/static/my_app/img/logo.jpg`
- ***SI TENEMOS VARIAS APLICACIONES DENTRO DEL MISMO PROYECTO Y TENEMOS CONTENIDO ESTÁTICO QUE SE LLAMA IGUAL***
- **Espacio de nombres de archivo estático**
 - Al igual que las plantillas, *podemos* considerar poner nuestros archivos estáticos directamente en `my_app/static` (en lugar de crear otro subdirectorio `my_app`), pero en realidad sería una mala idea.
 - ***Django elegirá el primer archivo estático que encuentre cuyo nombre coincida***, y tenemos un archivo estático con el mismo nombre en una aplicación *diferente*, Django sería incapaz de distinguir entre ellos.
 - Tenemos que ser capaces de indicarle a Django el correcto, y la forma más fácil de asegurar esto es ***organizándolos por espacio de nombres***. Es decir, poniendo los archivos estáticos dentro de *otro* directorio con el nombre de la aplicación.

Configuración

- A la hora de servir archivos estáticos el método (**`django.contrib.staticfiles.views.serve()`**) que utiliza es ineficiente para un entorno de producción.
- Este método sólo funciona cuando **DEBUG=True**
 - Es una variable de configuración (está en `settings.py`) que indica a Django que está en un entorno de desarrollo, cuando pasemos a un entorno de producción tendremos que dar una serie de pasos.

Archivos estáticos externos

- Dentro de nuestra aplicación podemos hacer **referencia a contenido estático** que no está vinculado a sólo una aplicación.
- A parte de utilizar la carpeta /STATIC/ se pueden **definir una serie de carpetas** que indican también contenido estático.
 - La variable se llamaría **STATICFILES_DIRS** y tendría que ir en el fichero de settings.py

```
STATICFILES_DIRS = [  
    os.path.join(BASE_DIR, "static"),           # Interna ...  
    '/var/www/static/',                         # Externa ...  
]
```

- *La variable **BASE_DIR** se configura en settings.py y apunta a la carpeta raíz del proyecto utiliza para ello el módulo os y la variable mágica **__file__** que muestra la ruta al fichero que se está ejecutando (**VER settings.py**)*

Enlaces

- <https://stackoverflow.com/questions/30430131/get-the-file-path-for-a-static-file-in-django-code>
- <https://stackoverflow.com/questions/11577681/a-simple-view-to-display-render-a-static-image-in-django>