


Open · Opened 2 days ago by  **Rubén Montero**

Un primer endpoint

Resumen

- Crearemos nuestra primera *app* dentro de `SimpleAPI` y la incluiremos en `settings.py`
- Añadiremos el fichero `endpoints.py`, que contendrá una función que sirve un *endpoint* sencillo
- Modificaremos `urls.py` para *mapear* una URL a dicha función, y servir así el *endpoint* a los navegadores

Descripción

Las convenciones de uso de Django indican que debemos [encapsular en apps](#) nuestras funcionalidades. Vamos a crear en `SimpleAPI` una nueva *app* llamada `simplerest04app`.

Para ello, usando un terminal (cmd.exe) debemos *posicionar* el directorio activo en la carpeta `SimpleAPI` que contiene `manage.py`.

Desde ahí:

```
python manage.py startapp simplerest04app
```

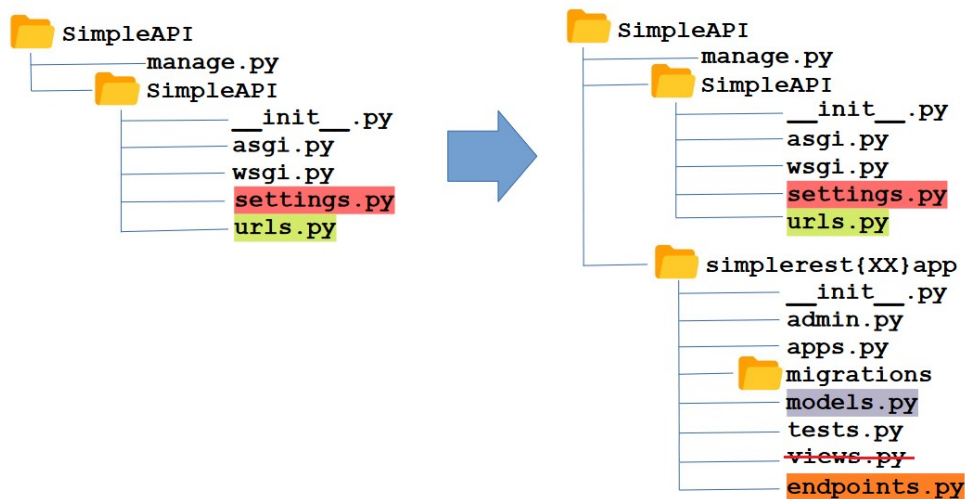
simplerest04app

Contendrá inicialmente una serie de ficheros. Ya conocemos algunos de ellos.

Comencemos **eliminando** `views.py`, desde el explorador de archivos o desde el IDE, mismamente.

Añadiremos un nuevo fichero `endpoints.py` que contendrá las *funciones* de *backend*.

Llegaremos a algo así:



settings.py

Como ya sabemos, para que nuestro proyecto [Django](#) (SimpleAPI) pueda hacer uso efectivo de nuestra *app* (simplerest04app), debemos añadir la siguiente línea a `settings.py`:

```
INSTALLED_APPS = [
+    'simplerest04app.apps.Simplerest04AppConfig',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
```

```
]

```

endpoints.py

Un fichero clave que contendrá el **código importante** de nuestro servidor REST.

De momento, añadamos una primera *función* que devuelva una sencilla respuesta en JSON:

```
from django.http import JsonResponse

def health_check(request):
    http_response = {"running": True}
    return JsonResponse(http_response)
```

(¿Recuerdas **qué** es el parámetro *request* ? ¡Lo usaremos próximamente!)

urls.py

Mientras dicha función no esté *conectada* a una URL ó ruta, **no sirve de nada**.

La segunda pieza clave que sirve a este propósito es el fichero `urls.py` :

Conectaremos nuestra función al *endpoint* `/health` , de tal manera que si visitas:

- <http://localhost:8000/health>

...verás la respuesta JSON deseada.

¿Cómo lo hacemos?

Así:

```
from django.contrib import admin
from django.urls import path
from simplereast04app import endpoints # ¡Acordémonos del import!

urlpatterns = [
    path('admin/', admin.site.urls),
    path('health', endpoints.health_check), # Esta línea mapea la URL a la función Python
]
```

Configuraciones en PyCharm

Recuerda que también puedes:

- Click derecho en la carpeta **SimpleAPI** padre (la que cuelga de `apis/`) > *Mark Directory as* > *Sources Root*, para prevenir errores de PyCharm interpretando los módulos disponibles

¡Listo!

Comprueba que ves el *endpoint* desde el navegador.

Por último

Verifica que tu código pasa el *test* asociado a la tarea.

Haz `commit` y `push` para subir los cambios al repositorio.



Rubén Montero @ruben.montero changed milestone to [%Sprint 4](#) 2 days ago



Ania Blanco @ania.blanco mentioned in commit [ca3110e1](#) 7 hours ago